IMPERIAL COLLEGE LONDON
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Structural Generative Descriptions for Temporal Data

Edgar Salomón García Treviño

March 2014

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy and the Diploma of Imperial College London

Declaration

I hereby certify that this submission is my own work, and that any idea or quotation from the work of others is acknowledged herein according to the standard referencing procedures.

Abstract

In data mining problems the representation or description of data plays a fundamental role, since it defines the set of essential properties for the extraction and characterisation of patterns. However, for the case of temporal data, such as time series and data streams, one outstanding issue when developing mining algorithms is finding an appropriate data description or representation.

In this thesis two novel domain-independent representation frameworks for temporal data suitable for off-line and online mining tasks are formulated.

First, a domain-independent temporal data representation framework based on a novel data description strategy which combines structural and statistical pattern recognition approaches is developed. The key idea here is to move the structural pattern recognition problem to the probability domain. This framework is composed of three general tasks: a) decomposing input temporal patterns into subpatterns in time or any other transformed domain (for instance, wavelet domain); b) mapping these subpatterns into the probability domain to find attributes of elemental probability subpatterns called primitives; and c) mining input temporal patterns according to the attributes of their corresponding probability domain subpatterns. This framework is referred to as Structural Generative Descriptions (SGDs).

Two off-line and two online algorithmic instantiations of the proposed SGDs framework are then formulated: i) For the off-line case, the first instantiation is based on the use of Discrete Wavelet Transform (DWT) and Wavelet Density Estimators (WDE), while the second algorithm includes DWT and Finite Gaussian Mixtures. ii) For the online case, the first instantiation relies on an online implementation of DWT and a recursive version of WDE (RWDE), whereas the second algorithm is based on a multi-resolution exponentially weighted moving average filter and RWDE. The empirical evaluation of proposed SGDs-based algorithms is performed in the context of time series classification, for off-line algorithms, and in the context of change detection and clustering, for online algorithms. For this purpose, synthetic and publicly available real-world data are used.

Additionally, a novel framework for multidimensional data stream evolution diagnosis incorporating RWDE into the context of Velocity Density Estimation (VDE) is formulated. Changes in streaming data and changes in their correlation structure are characterised by means of local and global evolution coefficients as well as by means of recursive correlation coefficients. The proposed VDE framework is evaluated using temperature data from the UK and air pollution data from Hong Kong.

Acknowledgements

I am indebt to Dr Javer Barria, my supervisor, for the academic opportunity he offered me to study my PhD at this renowned institution and for his guidance and support throughout the research process.

I gratefully acknowledge the financial support provided by CONACYT (National Council of Science and Technology of Mexico) which sponsored this work through a scholarship.

A special note of appreciation to Dr Vicente Alarcon Aquino for his encouragement to make me pursue an academic research career.

I am grateful to my family for always supporting me in whatever ventures I have chosen.

Thanks also to my friends from and outside Imperial for their constant support and fruitful discussions.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

1-NN	Nearest neighbour
ABS	Analysis by synthesis
ACF	Autocorrelation function
ARIMA	Autoregressive integrated moving average
ARMA	Autoregressive moving average
AWSOM	Arbitrary window stream modeling method
CC	Correlation coefficient
CH	Chebyshev distance
CO	Cosine distance
CR	Correlation distance
CUSUM	Cumulative sum control chart
CWRU	Case Western Reserve University
DCT	Discrete Cosine transform
DD	Detection delay
DFT	Discrete Fourier transform
DIKW	Data, information, knowledge and wisdom
DIK	Data, information and knowledge
DWT	Discrete wavelet transform
ECG	Electrocardiogram
ED	Euclidean distance
EDR	Electrodermal response
EEG	Electroencephalogram
EM	Expectation-maximisation
EWMA	Exponentially weighted moving average
EWMCC	Exponentially weighted moving correlation coefficient

FFT	Fast Fourier transform
FGM	Finite Gaussian mixtures
FIR	Finite impulse response
GEC	Global evolution coefficient
GLR	Generalised likelihood ratio
GMM	Gaussian mixture model
HMM	Hidden Markov model
IIR	Infinite impulse response
LEC	Local evolution coefficient
MFC	Mel-scale frequency cepstrum
MFCCs	Mel-scale frequency cepstral coefficients
MREWMA	Multiresolution exponentially weighted moving average
NFA	Number of false alarms
NN	Neural network
KDD	Knowledge discovery in databases
KDE	Kernel density estimator
ODAC	Online divisive agglomerative clustering
ODWT	Online discrete wavelet transform
OSE	Orthogonal series estimator
OSGD-D	Online structural generative description based on ODWT
OSGD-E	Online structural generative description based on MREWMA
OSGDs	Online structural generative descriptions
PAA	Piecewise aggregate approximation
PCA	Principal component analysis
RWDE	Recursive wavelet density estimator
SAX	Symbolic aggregate approximation
SE	Standardised Euclidean distance
SGDG	Structural generative description based on FGM
SGDs	Structural generative descriptions
SGDW	Structural generative description based on WDE
SNR	Signal to noise ratio
SVM	Support vector machine
SVP	Spatial velocity profile

TVP	Temporal velocity profile
UCR	University California Riverside
VDC	Velocity density estimation based on RWDE coefficients
VDD	Velocity density estimation based on RWDE densities
VDE	Velocity density estimation
VDKDE	Velocity density estimation based on KDE
WDE	Wavelet density estimator
WMA	Weighted moving average

Notation

ϕ	Scaling function
ψ	Wavelet function
$\tilde{\mathbf{H}}_m$	Scaling function filter for the m level of decomposition
$\tilde{\mathbf{G}}_m$	Wavelet function filter for the m level of decomposition
$< f, g >$	Inner product in the space $\mathbf{L}_2(\mathbb{R})$
$\|f\|$	Euclidean or Hilbert space norm
$\mathbf{L}_2(\mathbb{R})$	Space of square integrable functions
$\mathbf{x}^i$	The i input time series or data stream pattern from data set $\mathbf{X}$
$\bar{\mathbf{x}}^i_P$	The p time or wavelet domain subpattern for time series or data stream $\mathbf{x}^i$
$\breve{\mathbf{x}}^i_P$	The p probability domain subpattern for time series or data stream $\mathbf{x}^i$
$\mathbf{f}_{x^i}$	Feature vector for time series or data stream $\mathbf{x}^i$
n_ϕ	Order of the scaling function filter ϕ
2^{-m}	Resolution associated with level of decomposition m

Sets

$\mathbb{N}$	Positive integers including 0
$\mathbb{Z}$	Integers
$\mathbb{R}$	Real numbers

Probability

$\mathbb{E}(X)$	Expectation of random variable X
$\mathrm{Var}(X)$	Variance of random variable X

> Mathematics is the science of patterns. The mathematician seeks patterns in number, in space, in science, in computers, and in imagination. Mathematical theories explain the relations among patterns[...]. Applications of mathematics use these patterns to explain and predict natural phenomena that fit the patterns. Patterns suggest other patterns, often yielding patterns of patterns.
>
> LYNN A. STEEN,
> The science of patterns, Science 240(1988), 616.

Chapter 1

Introduction

The progress of hardware technology, which has included innovations in digital data acquisition, storage and communication, has laid the foundations for the enormous proliferation of databases in almost every area of human endeavour. The discipline concerned with the extraction of implicit and useful information from databases is known as *data mining.* Data mining is an interdisciplinary scientific discipline lying at the intersection of diverse research areas such as machine learning, statistics, pattern recognition and artificial intelligence.

Time series and data streams are two particular classes of temporal data objects which are pervasively important in a wide variety of fields ranging from science and engineering to business, finance and health care. However, traditional data mining techniques, such as Neural Networks (NNs) and Support Vector Machines (SVMs), were originally formulated in the context of so called *static data*, which involves static records with generally no predefined notion of time [2].

For the case of time series, data mining algorithms available generally follow one of the following two strategies [2]: 1) The first strategy is to modify data mining algorithms designed for static data in order to allow them to handle data whose features change over time. Here, the essential idea is to replace the distance or similarity measure used for static data with one suitable for time series; 2) The second strategy is to convert time series data into a static form and then directly apply mining techniques developed for static data. Unfortunately, data mining methods developed in the context of static data generally ignore the high amount of data dependency present in time series [3]. Furthermore, previously proposed time series mining solutions are based on the storage and analysis of fixed static data archives that allow complex mining operations based on multiple passes of the data.

Regarding data stream mining, existing approaches generally follow the same ideas as time series mining algorithms. However, the underlying characteristics of this particular class of data such as its unbounded size and fast arrival rate, prevent the direct application of traditional data mining methods to this domain. Hence, when analysing data streams, the research effort has been directed towards designing algorithms with reduced computational complexity as well as reduced memory requirements.

In this thesis the problem of time series and data stream mining is addressed by emphasizing the importance of data representation. To this end, concepts from the pattern recognition research community are used. According to the pattern recognition framework the data mining process can be divided into two equally important successive stages: data representation and data mining task algorithm (for example, classification, clustering, segmentation and change detection algorithms). Specifically, the data description stage focuses on the extraction of the set of properties or features essential for the characterisation of input data. This stage is crucial for the whole data mining process since it facilitates pattern extraction and it provides a reduction in dimensionality. Furthermore, from the pattern recognition perspective, two general types of data descriptions can be distinguished: 1) the statistical one, based on treating input patterns as single entities and then describing them using fixed-length feature vectors; and 2) the structural one, based on assuming input patterns as compound entities and representing them using variable-length structural attributes and relations. While the main advantage of statistical descriptions is the fixed-length representation of patterns, the main advantage of structural-based methods is the ability of modelling structural dependencies. By taking into account the above ideas, the framework proposed in this thesis is focused on designing a generic representation for temporal data that can capture structural dependencies into the form of a fixed-length feature vector. This representation can be easily combined with existing statistical or decision-theoretic techniques to construct novel data mining algorithms for temporal data.

Although the proposed framework is initially formulated considering an offline setting, its algorithmic instantiations are extended to work with both time series and data streams.

The second framework proposed in this thesis is particularly designed for analysing the evolution of multidimensional streaming data, where the term evolution refers to the process in which important changes occur over time in the underlying characteristics of the data stream. The proposed framework uses the probability density of the data to measure the rate of change of data concentration at a given spatial location, over a user-defined time horizon. In respect to previous approaches, this framework requires a significant lower amount of memory, is computationally less complex, and allows localised diagnosis at each dimension with only one pass of the data.

1.1 Motivation

In data mining problems the representation of input data plays a fundamental role, since it defines the set of essential properties for the extraction and characterisation of patterns [4]. However, for the case of temporal data, like for example time series and data streams, one of the major concerns regarding the developing of mining algorithms is finding an appropriate data representation [5, 6]. The novel frameworks and algorithms developed in this thesis are a step towards filling this gap in the data mining research activity.

Although data mining is a well consolidated and evolving research area in both academia as well as in industry, the offline and online frameworks and algorithms presented in this thesis are based on novel ideas that have not been previously investigated. Previous research has mainly focused on the data mining task [2] as well as on the learning algorithm [7], and the data description aspect has not been thoroughly addressed in the literature.

1.2 Aims & Objectives

The aim of the work presented in this thesis is to develop novel data representations for time series and data streams useful to perform offline and online data mining tasks. The thesis describes the effort to accomplish this goal by systematically addressing the following objectives:

- To formalise a domain-independent description framework for temporal data referred to as Structural Generative Descriptions (SGDs) based on a pattern description strategy that combines structural and statistical pattern recognition approaches in a novel fashion.
- To formulate offline algorithmic instantiations of the proposed SGDs framework useful in the context of time series mining.
- To formulate online algorithmic instantiations of the proposed SGDs framework suitable for the mining of data streams.
- To formulate a multidimensional data stream evolution diagnosis framework which incorporates a Recursive Wavelet Density Estimators (RWDEs) and adapts the concept of Velocity Density Estimation (VDE).

1.3 Highlights of Main Contributions

The main contributions of this thesis are the following:

- Moving the structural pattern recognition problem to the probability domain, and formulating a novel statistical-structural representation for temporal data.
- Formulation of a novel time series representation framework useful for data mining applications.
- Formulation of two offline and two online algorithmic instantiations of the proposed SGDs framework, useful for time series and data streams mining, respectively.
- Formulation of a novel evolution diagnosis framework based on RWDE algorithms.

1.4 Organisation of the Thesis

This thesis is organised in three major parts. The first part, which includes Chapter 1 and Chapter 2, presents a general overview of the outstanding representation issues in data mining tasks involving temporal data (Chapter 1); as well as a brief review of background theories relevant to the algorithms and solutions proposed in this thesis (Chapter 2), which include theoretical aspects of pattern recognition, machine learning, sparse signal processing, nonparametric statistics as well as KDD and data mining.

The second part of this thesis, comprised of Chapter 3 and Chapter 4, presents the proposed SGDs time series representation framework as well as two offline and two online algorithmic instantiations. While Chapter 3 introduces the framework and algorithms, Chapter 4 focuses on its corresponding empirical evaluation.

The third part of this thesis, entirely contained in Chapter 5, presents both the formulation of a novel RWDE-based data stream evolution framework and its corresponding experimental evaluation.

The thesis ends in Chapter 6, with the conclusions and the recommended research work for future researchers.

In Figure 1.1 a block diagram relating the above described arrangement of the thesis is presented. This diagram includes the interconnections between different chapters.

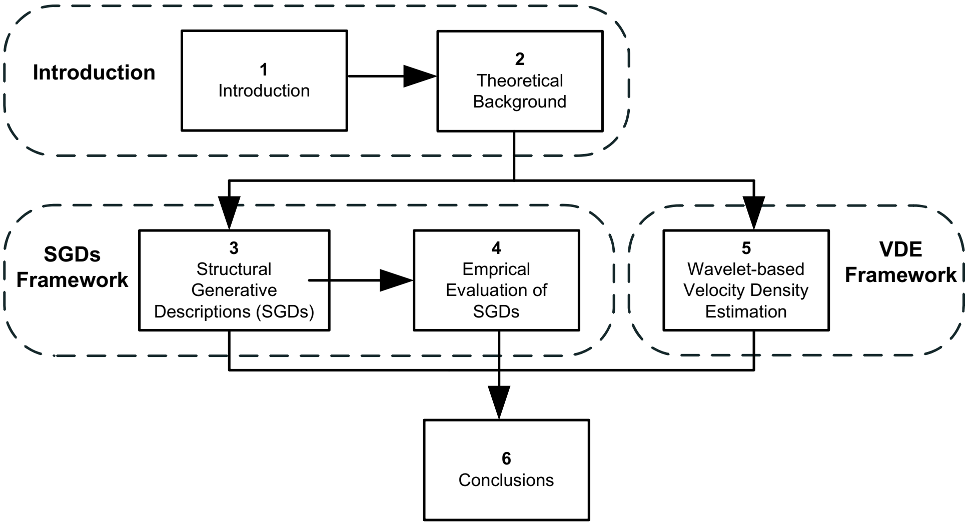


FIGURE 1.1: Arrangement of the chapters of the thesis.

1.5 Thesis Synopsis

In this thesis different frameworks and algorithms are proposed to provide novel representations for both time series and data streams. The core of the thesis is the SGDs framework whose essential idea is moving the structural pattern recognition problem to the probability domain and in this way, combine the two main pattern recognition paradigms. The proposed SGDs framework is motivated by the observation that in pattern recognition problems involving complex patterns or problems in which the structural dependency is important, an effective strategy would be to describe each pattern in terms of simpler subpatterns and the relations among them [8]. The SGDs framework involves two main stages: 1) decomposing time series into simpler subpatterns in time or any other transformed domain, and 2) mapping of these subpatterns to the probability domain to obtain probability domain subpatterns. Here, each probability domain subpattern is a structure, that in turn, can be divided into simpler elements or primitives. In this way, the proposed SGDs-based representation strategy is based on describing or representing input time series by the set of attributes and relations of the primitives associated with their corresponding probability domain subpatterns. Two algorithmic instantiations of the SGDs framework, suitable for time series classification, are developed considering multiresolution approximation concepts as well as nonparametric density estimation.

The SGDs time series representation framework is then extended to the online context to make it suitable for data stream applications. Two distinct online algorithms based on SGDs concepts are developed. The algorithms are based on fast online formulations of both multiresolution decomposition and density estimation blocks. In addition to combining statistical and structural pattern recognition paradigms, the algorithms also fulfil some of the basic requirements for data stream mining algorithms: 1) fast and incremental data processing; 2) constant computational complexity and fixed amount of memory required, 3) compact representations of data streams at each time stamp, 4) both, concept shift and concept drift detection are possible.

In the streaming context, data evolution refers to the process in which important changes occur over time in the trends of a given data stream due to changes in the underlying phenomena [9]. Although a plethora of algorithms have been designed for change detection in streaming data [10–13], regarding the online characterisation of its evolution, the most relevant work related to the algorithm developed in this thesis is the concept of *Velocity Density* [9] which involves measuring the rate of change of data concentration at a given spatial location over a user-defined time horizon. In this thesis, the diagnosis of the evolution of multidimensional streaming data is performed using a framework that adopts the concept of velocity density and incorporates RWDE algorithms. In the proposed framework changes in streaming data are characterised by the use of local and global evolution coefficients. In addition to this, for the analysis of changes in the correlation structure of the data the recursive estimation of correlation coefficients is proposed.

> From now on, the key is knowledge. The world is not becoming labor intensive, not material intensive, not energy intensive, but knowledge intensive.
>
> Peter F. Drucker
>
> Managing for the future: The 1990s and beyond. Dutton Adult, 1992

Chapter 2

Theoretical Background

This chapter provides a general overview of the theoretical background for the thesis. Relevant theoretical concepts from the fields of pattern recognition, machine learning, sparse signal processing, nonparametric statistics as well as Knowledge Discovery in Databases (KDD) and data mining, are reviewed.

The chapter is organised in five sections. In Section 2.1 a brief introduction of the relevance and theoretical connections of the frameworks reviewed in this chapter with the work presented in this thesis is provided. In Section 2.2 the pattern recognition paradigms are reviewed. In Section 2.3, the three main approaches for machine learning are presented. Section 2.4, focuses on reviewing the so called KDD and data mining frameworks. Finally, Section 2.5 and Section 2.6, are dedicated to provide theoretical details about the sparse signal processing and nonparametric statistics research fields.

2.1 Introduction

As Duin *et al.* emphasised in their review paper about pattern recognition achievements and perspectives [14], in science new knowledge is generally formulated in terms of existing knowledge. In that sense, researchers can only achieve what is derived from the corresponding assumptions and constraints of the particular framework in which they stand. This remark underpins the investigations that led to the time series and data streams representation frameworks and algorithms proposed in this thesis. The most relevant scientific disciplines that influenced and inspired this work are: pattern recognition, machine learning, KDD and data mining, sparse signal processing as well as nonparametric statistics (see Figure 2.1 for a block diagram). All these frameworks are reviewed in this chapter.

FIGURE 2.1: Frameworks related to the work presented in this thesis.

2.2 The Pattern Recognition Framework

The first and foremost influence for the proposed SGDs framework and its corresponding offline and online algorithms presented in Chapter 3, are methods combining structural and statistical pattern recognition paradigms, from which the concept of structural descriptions with statistical classification comes from.

Pattern recognition is a field of study that focuses on how machines can distinguish patterns in their environment and make decisions about the categories of the patterns [15]. In that sense, the fundamental goal in pattern recognition is supervised or unsupervised classification [15]. Given this goal, the functionality of pattern recognition systems can be divided into two main successive tasks: description and classification. The first of these tasks, the *description* task, extracts the set of properties or features required for the characterisation of patterns, while the *classification* task involves mapping the set of extracted features to a particular group or class. In Figure 2.2 the corresponding block diagram for the pattern recognition process is shown.

Since the beginning of 1960s, time at which pattern recognition emerged as a new topic for research, many mathematical methods have been proposed for solving pattern recognition issues. Two major paradigms for implementing pattern recognition systems can be distinguished in all those techniques [16]: *statistical* and *structural*.

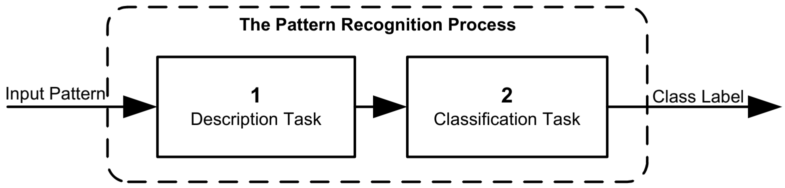


Figure 2.2: The pattern recognition framework.

In the *statistical* approach, also known as *decision-theoretic* approach, each input pattern is treated as a single entity and then is described or represented by a fixed n-dimensional vector of numerical features. In this way, the classification task involves the partition of the feature space into regions, each of them associated with a single class [8]. Typically, this is accomplished by applying firmly established techniques from discriminant analysis framework or statistical decision theory.

In the *structural* approach, also referred to as syntactic pattern recognition due to its origins in formal language theory [16], each pattern is treated as a combination of multiple entities and then is described or represented by simpler subpatterns and their structural or topological relations. In this context, the simplest or elemental subpatterns in which the input pattern can be decomposed are usually called *pattern primitives* or simply *primitives.* Syntactic approaches specifically considers the analogy between the structure of patterns and the syntax of a language [17]. In that sense, primitives and input patterns are viewed as the alphabet and the sentences, respectively [15]. The classification task is performed, in those techniques, by matching or parsing the variable-length structural representation of the pattern according to a set of syntax rules or grammars [18].

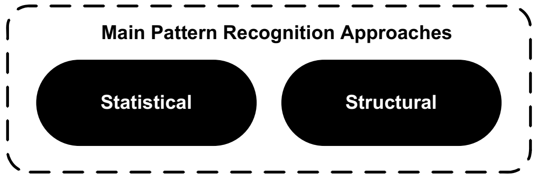


Figure 2.3: Main pattern recognition approaches.

Both paradigms of pattern recognition have their corresponding advantages and disadvantages. Statistical approaches are, to some extent, capable of handling pattern deformation due to noise and distortion, because they are founded on numerical feature data [18]. Additionally, they allow the application, in the classification stage, of well-established statistical and decision-theoretic techniques [18]. However, a key issue inherent to these kind of approaches is the lack of description they offer regarding the pattern to be recognised [4] and the incapability to handle variable-length descriptions. Note here that a set of features may be useful for the categorisation of patterns into different groups; however, it does not necessarily provide information related to their structure. In addition to this, since statistical approaches rely on numerical features, when the number of classes is very large or when the complexity of the patterns under study is very high, the number of features required for the corresponding patterns characterisation also becomes very large [8]. In machine learning, this phenomenon is usually referred to as the *curse of dimensionality* [19]. In general, in techniques belonging to this group, greater emphasis is given to the classification stage rather than to the description phase [4].

For the case of structural approaches, their main advantage is the fact that they are not only able to classify variable-length features, but also that they provide appropriate pattern descriptions. However, there is not a generic solution for the selection or extraction of pattern primitives, and it is difficult to construct syntactic grammars that embody a precise criteria to differentiate among different groups. With regard to the latter issue, grammars are either too simplistic, in order to be applied in different domains, or too complex and, in that sense, requiring additional domain knowledge [5]. Furthermore, structural approaches are, in their basic formulation, incapable to handle distortions and noise [18].

2.2.1 Combined Statistical and Structural Models

A few years after the establishment of the pattern recognition framework it was noticed by the research community that statistical and structural approaches could be complementary [16]. Since then, multiple efforts have been dedicated to develop combined approaches that allow taking advantage of the complementary capabilities and strengths of both paradigms and then offer general solutions for most pattern recognition applications [18]. The literature related to such combined models is quite substantial, however, most of those research efforts follow one of the following two strategies.

2.2.1.1 Statistical into Structural

The first strategy is the inclusion of statistical information into structural approaches (see, for example reviews in [20] and [18]). Approaches following this strategy can be categorised into two main groups. Methods within the first category rely on the inclusion of statistical information into primitives via *error correction transformation/matching* or into subpattern structures via *stochastic production rules.* Methods following the concept of error transformation basically assume that in a string, a symbol can be transformed into another according to a set of possible operations. In [21] a Bayes-based pattern deformational model which considered a separate treatment of primitive syntactic information (describing primitive structure) and semantic one (describing primitive properties) was investigated. Stochastic production rules-based approaches rely on the introduction of probabilistic aspects into the syntactic model by considering probabilistic structure variations of subpatterns. In this context, [22] studied a parsing algorithm that provided probabilistic membership criteria for strings recognition. The second group of techniques incorporating statistical information into pattern structures include approaches relying on *primitive generalisations* which are introduced in primitives and subpatterns representations via *attributed primitives* [18]. In general terms, attributes are a set of numerical values that specify other characteristics of the primitives. Grammars considering attributed primitives are referred in the literature as *attributed grammars* and they were introduced in order to enable the computation of subpattern attributes during the parsing process. In [23] is reported a system in which semantic rules of attributed grammars were used to guide extraction of primitive and subpatterns attributes. In this context, the attempt presented in [8] theoretically formalises the concept of attributed grammars as the initial step toward the unification of syntactic and statistical pattern recognition approaches. A relevant paper combining statistical/structural models is [24] where geometrical-statistical shape descriptions are integrated into a handwritten characters recognition framework in order to enhance robustness against shape deformation. The method specifically considers two types of features namely, quasi-topological (which are symbolic, qualitative and discrete) and geometrical (which are numerical, quantitative and continuous). In this case the classification stage is performed through structural matching algorithms.

Here is important to note that, although the proposed SGDs framework, presented in Chapter 3, is mainly based on expressing structural aspects in a statistical form, it also incorporates the idea of attributed primitives. In this sense, the primitives selected in the proposed SGDs framework have associated some numerical attributes that modify the way they are combined to construct the corresponding subpatterns.

2.2.1.2 Structural into Statistical

The second strategy is based on including structural information into statistical approaches. Here, just few authors have reported on this approach in the technical literature. In [25], a procedure based on the combination of both structural and statistical features was proposed for handwritten character recognition. Specifically, character structural features, that is, strokes, intersections, holes, and extremes, were mapped into a fixed-length feature vector by a parameterising strategy that obtains continuous numerical values calculated in relation to the center of gravity of the pattern. The classification was performed applying statistical classification based on a linear discriminant-based classifier. In the context of optical character recognition, the approach described in [26] uses five shape primitives, that is, *stroke*, *hole*, *arc*, *crossing* and *endpoint* to represent the structure of each character, and then constructs a function called *feature identification mapping* that maps structural representations of characters into binary feature vectors. These feature vectors are subsequently processed using a statistical classification technique relying on kD-trees. Statistical and structural approaches are combined in [27] for fingerprint image postprocessing. Minutia structures are represented based on fingerprint ridges attributes calculated around each minutia using minutia distance as statistical measure. In [4], a character description method is presented, based on the evaluation of geometrical moments on structural representations of the characters in terms of *circular arcs* primitives. Since, geometrical moments constitute a fixed-length feature vector a statistical classifier, that is, neural network, is employed for the classification stage.

It is important to highlight that the proposed SGDs framework presented in Chapter 3 can be categorised within this group of combined pattern recognition approaches, since it is based on incorporating some structural characteristics of the time series into a fixed-length feature vector. Note however that, in the proposed SGDs framework, the structure of the time series is analysed in the probability domain.

2.2.2 The Analysis by Synthesis

The novel SGDs time series representations proposed in Chapter 3, which considers a module able to generate patterns similar to the input pattern, are particularly related to the *analysis-by-synthesis* (ABS) speech recognition model introduced by Halle and Stevens [28]. This model was formalised a few years before the syntactic pattern recognition framework was firmly established. Note that according to Kanal [16], syntactic pattern recognition is reminiscent of ABS.

The idea behind the ABS approach is to explain observed patterns in terms of a compact set of hidden causes that generate them [29]. Specifically, the analysis-by-synthesis model, in its initial formulation, assumes the generation of synthetic patterns and their corresponding matching with input patterns. The emphasis is made on the use of a generative synthesiser model that embodies the physical process governing input patterns generation and its parameters adjustment using such input patterns. The characterisation and recognition of a given pattern is then performed based on the set of parameters that provides the best match between synthesised and input parameters.

In its initial formulation, the ABS recognition model relies on the mapping from the pattern to the class space through an active feedback process [28]. Specifically, patterns are internally generated in an analyser/synthesiser according to a flexible or adaptable sequence of instructions, that is, a parameter adjustment/learning process, until a best match with the input pattern is obtained. The recognition of patterns is performed by examining the corresponding internal configuration that is, the corresponding parameters, of the analyser/synthesiser for each pattern.

By considering that the term *structural pattern recognition* includes all those approaches based on defining primitives and identifying allowable structures in terms of the relations among primitives and substructures, it can be generalised that in broad terms structural pattern recognition assumes that patterns are constructed or "generated" according to all these elements, that is, according to their primitives, subpatterns and relations [16]. In that sense, there is a generative process or mechanism underlying input patterns descriptions, and then a fundamental connection between structural and ABS pattern recognition frameworks. This connection becomes more evident when the emphasis is no longer placed on identically matching input patterns or physical processes, but rather on constructing black box generative models that generate patterns with similar properties than the input patterns.

2.3 The Machine Learning Framework

The idea of including generative aspects into the SGDs framework and algorithms of Chapter 3 is taken from two main sources, the analysis-by-synthesis approach [28] (reviewed in Section 2.2.2) and the hybrid machine learning framework discussed in [7]. The connection with the latter framework comes from realising that the proposed generative descriptor can be seen, within the machine learning framework, as a generative learning process.

Machine learning focuses on building computer systems able to adapt to their environment and learn from their experience [30]. According to this framework, supervised classification methods can be organised into two main groups: generative and discriminative [7, 31]. On the one hand, generative approaches involve learning a model that describes the input/feature space based on the understanding of the causality between groups or classes and their corresponding observed input/features. On the other hand, discriminative methods imply the optimisation of a decision rule that organise data into different categories, without considering the causal relationships between input/feature data and the underlying generative process.

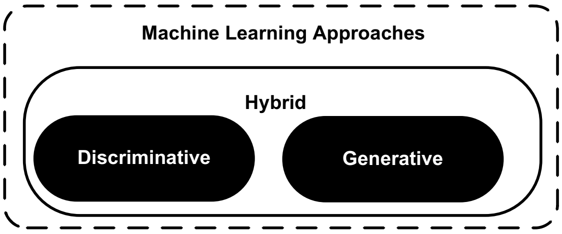


Figure 2.4: Machine learning approaches.

Particularly, generative approaches involve two steps: first, the construction of a model representing the joint distribution of the input features and the output labels for different classes, and second, the formulation of a decision rule able to distinguish between those categories using the constructed model. With respect to discriminative techniques, their main objective is the direct finding of a decision rule that distinguishes between different classes [32]. Figure 2.5 pictorially shows the two learning approaches.

In a formal way, using the random variables X and C to represent input features and their corresponding class labels, and assuming that $X \in \mathbb{R}$ and $C \in \mathbb{N}$ with C taking only a fixed number of finite values corresponding to different classes. Generative classification intends to learn the joint distribution $P(X, C)$ by the use of parametric or nonparametric models, and then apply Bayes rule to compute the posterior conditional distribution $P(C|X)$ to assign the most likely label c for each new data vector $\mathbf{x}$. In that sense, in generative approaches, the classification task is effectively reduced to modelling the distributions $P(X, C)$ and $P(C)$ for each class c. According to this, the decision process is based on finding the class c with the highest probability of generation of the observation represented by the input feature vector $\mathbf{x}$. In counterpart, discriminative classification involves the direct learning of the classification rule defined by the posterior distribution $P(C|X)$ without the initial estimation of the joint distribution $P(X, C)$. For that reason, discriminative techniques can be interpreted as function fitting-based methods whose objective is to learn a direct mapping from the input X to the output label C. This mapping can be done either by the approximation of the conditional probability distribution $P(C|X)$ or through other methods that achieve minimal classification error.

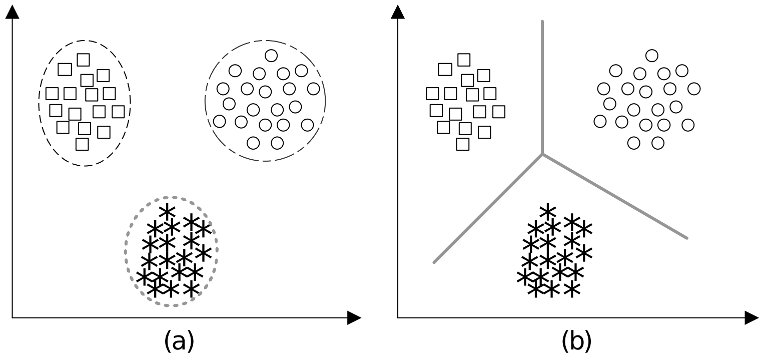


Figure 2.5: Machine learning approaches [33]: (a) Generative; (b) Discriminative.

2.3.1 Discriminative versus Generative Techniques

Both discriminative and generative approaches have their own advantages and disadvantages. Discriminative methods are generally believed to be superior to its generative counterpart, since they focus their attention on finding a decision boundary that separates different classes, and in that sense, they directly optimise the quantity of interest, that is, the classification error [32]. In counterpart, general generative models are usually less optimised for classification task because they indirectly optimise the classification error by learning descriptions of classes separately through their joint distributions [32, 34].

An important advantage of generative models is that, since an independent model is learned for each class, they allow modular learning. As a consequence, there is a simplification in the learning process because no interaction between models of different classes is considered. Moreover, in this context, addition or subtraction of classes is possible and straightforward. In contrast, the learning process of discriminative models is a global one since a single model is learned for all classes, making difficult or impossible the addition or subtraction of classes [34, 35]. Note that the above concepts not only apply for binary and multi-class classification problems both also for one class classifiers. Regarding one class classifiers the distinction between the two machine learning approaches is subtler since the task is reduced to finding a model that fits the data. In this case, generative approaches construct a generation model for all the data, while discriminative approaches focus on optimising only their frontier.

Furthermore, generative models capture the underlying generation process of a data population of interest, for that reason, they offer more insights about the structure of input data [32] and allow the incorporation of prior knowledge about the domain [7]. On the contrary, discriminative models possess a limited modelling capability, since they are focused on classification boundaries rather than the generation process of the data [36]; in that sense, they generally ignore the rest of the space and they offer no insights about the structure of input/feature data [32]. For this reason, discriminative approaches are usually hard to interpret, because they are founded on the treatment of classes-feature relations as a black box [32]. Finally, discriminative methods require more training data for the parameters to converge [32], while generative models are capable of learning even in the presence of some missing values [34].

The complementarity of generative and discriminative learning has motivated a number of authors to seek hybrid methods in order to combine their strengths [37]. Additionally, as data become more complex and high-dimensional, it is clear that a single method is not sufficient to fulfil the exigencies of modern classification-based applications. In [32] it has been suggested that an optimal classification strategy should include, firstly a generative model to deal with missing and few amount of training data, and secondly, a discriminative model in order to converge to a model with lower asymptotic error when sufficient data is available for discriminative learning.

2.3.2 Hybrid Machine Learning Approaches

Recently, a third category of classification techniques has been the focus of interest of the research community, it includes all those algorithms referred in the literature as *hybrid* approaches, whose fundamental idea is the incorporation of multiple classifiers, from the previously explained generative of discriminative categories, into a single and more robust classifier. The fundamental concept behind hybridisation is the fact that the combination of classifiers overcomes deficiencies caused by the use of one particular algorithm. In that sense, hybridisation allows exploiting the advantages of multiple classification approaches while overcoming their weaknesses [38]. Relevant work involving hybridisation concepts is the two stage classifier that combines Hidden Markov Models (HMM) and Support Vector Machines (SVMs) introduced by [34] in the context of time series and sequence classification. In the first stage, referred as modelling stage, p-HMMs are used, in a p-class problem, to map time series data into a fixed p-dimensional vector containing likelihood scores which are indicators of how likely the model has generated a particular time series or sequence. In the second stage, a SVM algorithm is used to classify time series data according to corresponding likelihood scores. In [39], the authors proposed a two phase generative-discriminative algorithm for visual categorisation based on the use of Fisher Kernels, which are the derivative of the log likelihood of the parameters, as features for classification. In the generative phase of the algorithm constellation models, trained using the EM algorithm [40] are used to construct probabilistic models of object classes by representing the appearance and relative position of object parts. Fisher Kernels are calculated from the constellation models and then, in the second phase of the classifier, they are used by a SVM algorithm to perform the final classification. For supervised scene classification, in [41] a two stage generative-discriminative classifier is investigated following the concept of dimensionality reduction via latent generative models to improve classification performance. Specifically, a probabilistic latent semantic analysis, a generative model from the statistical text literature, is used in the first stage as a statistical clustering method to discover latent topics. The idea here is the determination of the model with the highest probability of generation of the distribution of "visual words" that appear in a given image. In the second stage, the topic distribution vector of each image is used as a feature to train a KNN and SVM multiclass discriminative classifier.

It is important to note that, from the machine learning perspective, the SGDs representations proposed in Chapter 3 can be seen as a generative block, since it is based on constructing a probabilistic model that describes the data at different resolutions. This generative block, when combined with a discriminative classifier, will give rise to a novel hybrid generative-discriminative algorithm.

2.4 The Knowledge Discovery in Databases (KDD) and the Data Mining Frameworks

Before starting to review some relevant theoretical fundamentals of KDD and Data mining frameworks it is important to clarify the concepts of data, information and knowledge according to the way they are used and understood in this thesis.

2.4.1 Data, Information and Knowledge

Data, information and knowledge are some of the fundamental concepts of this era. These three polyvalent concepts have been the focus of intense debate among experts from information theory, artificial intelligence and KDD. In Table 2.1 some of their highly accepted definitions are presented.

Table 2.1: Three definitions for data, information and knowledge.

Data	**Information**	**Knowledge**
• Discrete, objective facts and observations, which are unorganised and unprocessed, and do not convey any specific meaning [42].	• Data shaped into a form that is meaningful and useful to human beings [43].	• Information put to productive use [44].
• Elementary and recorded descriptions of things, events, activities and transactions [43].	• Data processed for a purpose [45]	• Information combined with understanding and capability [43].
• Symbols representing properties of objects, events and their environment [46].	• Data given a meaning by way of context [47].	• Structured and organized information that has developed inside a cognitive system [48].

From Table 2.1 it is clear that although there are no universally accepted definitions for data information and knowledge, the implicit assumption is that information is extracted from data whereas knowledge is build up from information. In order to explore the process associated with the transformation of these three concepts, in [49], Ackoff introduced the so called Data-Information-Knowledge-Wisdom (DIKW) hierarchy which is usually depicted in the form of a pyramid, with data at the base and wisdom at the top. Since in the context of KDD only the first three concepts are relevant, the DIKW hierarchy can be reduced to the Data-Information-Knowledge (DIK) pyramid shown in 2.6.

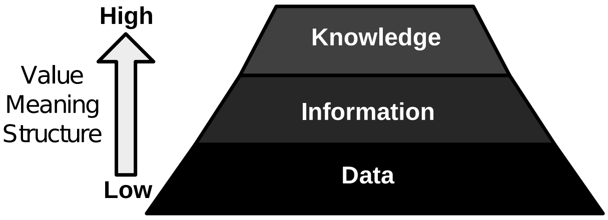


Figure 2.6: The Data-Information-Knowledge pyramid [46].

Note that in the DIK pyramid of Figure 2.6 a movement up in the pyramid involves a qualitative refining process associated with an increase in meaning, structure and understanding. Note also that according to this model, there is not information without data and, in turn, there is not knowledge without information.

Since the work presented in this thesis is aimed at the construction of representations of time series and data streams, it can be located between the data and information blocks of the pyramid of Figure 2.6.

2.4.2 The Knowledge Discovery in Databases Framework

The information age, is characterised by the speed and ubiquity of data and information. According to Hilbert *et al.* [50], in 2007, humankind was able to store around 2.9×10^{20} bytes and to communicate (via TV, radio, Internet, telephone, etc) around 2×10^{21} bytes. This is evidence that today the amount of data being generated exceeds the human ability to perform manual analysis and hence there is a crucial and urgent need for tools and techniques for intelligent data processing and understanding.

The term *Knowledge Discovery in Databases (KDD)* was coined in 1989 by Gregory Piatetsky-Shapiro in the workshop held in Detroit, USA, under the same name [51]. Since then, the area has been subject of increasingly intensive research, attracting the attention of people and scientists from most diverse fields and disciplines.

In this thesis the classical definition of KDD is followed. According to [52], KDD is the non trivial iterative and interactive process which focuses on identifying valid, novel, potentially useful and understandable patterns in data. The fundamental goal of the whole KDD process is the extraction and application of the knowledge derived from the patterns extracted from data.

Note that since KDD denotes the overall process of extraction of high-level knowledge from low-level data [53] its first step is the selection of the application domain as well as the definition of the final goal of the knowledge discovery. Once the application domain and the goal of the KDD process have been identified, the KDD process can be subdivided into the sequence of iterative steps or stages shown in Figure 2.7 [53].

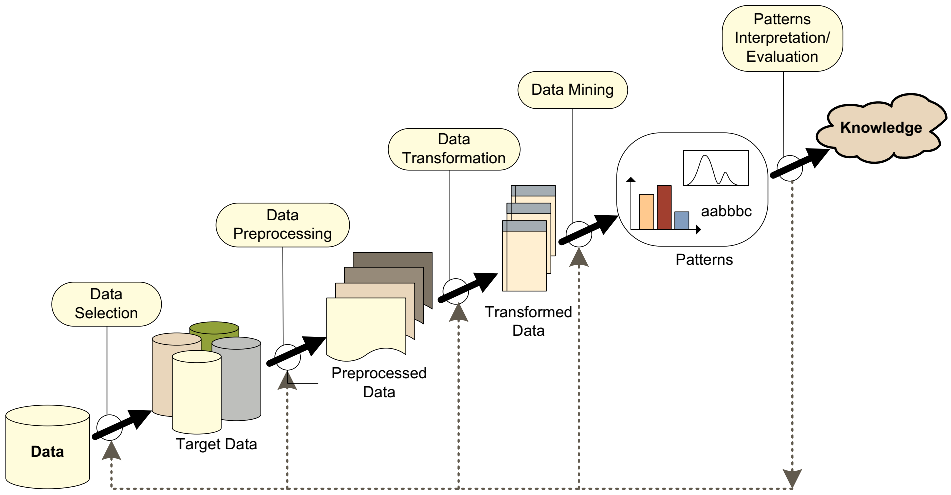


Figure 2.7: The KDD process [52].

1. **Data Selection**, which, by taking into account considerations about the homogeneity of the data, focuses on the selection of a subset of data samples over which the discovery will be performed.

2. **Data Preprocessing**, which includes operations such as data cleaning, data normalisation as well as noise and outliers removal.

3. **Data Transformation**, which involves reducing the number of variables under consideration and also considers the process of finding useful features to represent data.

4. **Data Mining**, which is aimed at the extraction of patterns of interest from transformed data.

5. **Patterns Interpretation/Evaluation**, which is oriented towards the interpretation and visualisation of the extracted patterns as well as its corresponding understanding in the context of the application domain.

2.4.3 Data Mining

The core of the KDD process is the data mining stage which refers to the formal study of methods and algorithms for the extraction of implicit and useful information from data [54]. In the data mining framework, raw data or simply data are characterized as recorded facts while information is defined as the set of patterns underlying data.

Data mining is an interdisciplinary subfield of computer science involving methods lying at the intersection of artificial intelligence, machine learning, statistics, and database systems. According to the task they address, data mining algorithms can be categorised into the following groups [30]:

1. **Exploratory Data Analysis**. It focuses on exploring the data in an interactive and visual way to find patterns that may seem interesting to the user. Since it is an interactive process involving the user, it posses a subjective element.

2. **Descriptive Modeling**. The goal here is to construct a model that describes the data or the process generating the data. The main tasks included within this category are: summarisation, density estimation, clustering, segmentation, dependency modelling.

3. **Predictive Modeling**. The idea here is, using the known values or categories of a variable, build up a model able to predict the category or value of unknown variables. Within this category four main tasks can be distinguished: classification, regression, prediction, and anomaly detection.

4. **Discovering Patterns and Rules**. It is aimed at finding combinations of items that appear frequently in a data set

5. **Retrieval by Content**. The objective is to search for patterns similar to the ones provided by the user. It is commonly used in the context of text and image data.

2.4.4 Data Representation Stage of the Data Mining Block

In data mining algorithms, prior to the data mining task there is usually a feature extraction or feature reduction block, in which the main goal is the extraction of characteristic quantities or properties from the data that serve as inputs for the data mining task. According to the KDD process of Figure 2.7 this block may be included within the data transformation step. However, in many applications (including the work reported in this thesis) the attention is focused on the extraction of patterns rather than the discovery of knowledge, and hence the data mining step is usually detached from the KDD process and employed separately. Therefore, in the work reported in this thesis the feature extraction block could also be seen as part of the data mining step.

In this thesis, the generic name *data representation* is used to group all those substages oriented towards the construction or extraction of a meaningful representation of input data. Then, according to their functionality, and taken into account concepts from pattern recognition reviewed in Section 2.2, two main stages in data mining algorithms can be recognised: 1) the data representation block and 2) the data mining task itself (for example, classification, clustering and segmentation.). See Figure 2.8 for a block diagram.

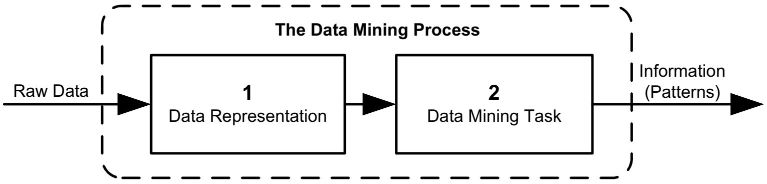


FIGURE 2.8: Functionality of the data mining process using pattern recognition concepts.

Note that although there is voluminous literature regarding the data mining framework, many of the existing works are targeted at studying the data mining task rather than focusing on the way data are described or represented prior to the application of a given data mining algorithm. In this thesis the emphasis is specifically placed on the representation stage, and in this sense, the frameworks and algorithms proposed in this thesis are aimed at the construction of time series and data streams representations that facilitate and improve the mining task.

Note that the rationale guiding the investigations proposed is the fact that the performance of data mining algorithms directly depends on the richness of information contained in input data, then superior levels of performance in the extraction of patterns from data can only be reached by combining both robust mining algorithms with rich data representations.

2.5 Sparse Signal Processing

One of the key aspects of the SGDs framework and algorithms developed in this thesis is the concept of multiresolution decomposition of data which has been taken from wavelet theory. Wavelet-based analysis comprises a group of methods that fall within the so called sparse signal processing [55] which focuses on the study of operators that provide *sparse representations* of signals. In this context, sparsity is related to the property of using a few number of coefficients to represent a given signal. Sparse signal processing includes a wide range of techniques such as Fourier, Wavelet and Cosine transforms, Empirical Mode Decomposition [56] as well as Frames [57, 58]. A succinct and general review of the fundamental concepts behind wavelet theory, is presented in this section.

2.5.1 Wavelets and Multiresolution Approximations

Wavelet analysis is a well-established discipline whose basic concept is the projection of data onto a set of basis functions in order to separate different scale information. Particularly, in the Discrete Wavelet Transform (DWT) data is separated into detail coefficients (fine-scale information) and approximation coefficients (large-scale information) by the projection of the data onto an orthogonal dyadic basis system [59]. The Discrete Wavelet Transform (DWT) is from the practical point of view the most important algorithm in wavelet theory [60], since it is easy to implement and it reduces the computation time and resources required. *Multiresolution signal approximations* are the theoretical foundations for such transform.

The basic idea behind the concept of multiresolution approximations is the approximation of signals at different level of resolutions by the use of orthogonal projections on different spaces $\{\mathbf{V}_m\}_{m\in\mathbb{Z}}$. In order to compute these projections an orthonormal basis for $\mathbf{V}_m$ is required. Specifically, a multiresolution approximation is a sequence of nested closed subspaces $\{\mathbf{V}_m\}_{m\in\mathbb{Z}}$ in $\mathbf{L}^2(\mathbb{R})$ that provides a formal approach for the construction of scaling orthonormal bases. The formal definition for these particular types of approximations is presented in Definition 1.

Definition 1. **Multiresolution Approximation**: A sequence $\{\mathbf{V}_m\}_{m\in\mathbb{Z}}$ of closed subspaces of $\mathbf{L}^2(\mathbb{R})$ is a multiresolution approximation if the following mathematical properties are satisfied [55]:

1. **Nested/Causality:** $\forall m \in \mathbb{Z}, \quad \mathbf{V}_{m+1} \subset \mathbf{V}_m$;
2. **Density:** $\lim_{m\to-\infty} \mathbf{V}_m = \text{Closure}\,\{\cup_{m\in\mathbb{Z}}\mathbf{V}_m\} = \mathbf{L}^2(\mathbb{R})$;
3. **Separation:** $\lim_{m\to\infty} \mathbf{V}_m = \{\cap_{m\in\mathbb{Z}}\mathbf{V}_m\} = \{0\}$;
4. **Scaling:** $\forall m \in \mathbb{Z}, \quad f(x) \in \mathbf{V}_m \leftrightarrow f(\frac{x}{2}) \in \mathbf{V}_{m+1}$;
5. **Translation:** $\forall m, l \in \mathbb{Z}, \quad f(x) \in \mathbf{V}_m \leftrightarrow f(x - 2^m l) \in \mathbf{V}_m$;
6. **Orthonormal Basis:** $\exists \phi \in \mathbf{V}_0 \mid \{\phi(x-k)\}_{k\in\mathbb{Z}}$ is an orthonormal basis for $\mathbf{V}_0$.

The first property is related to the nested arrangement of the subspaces $\mathbf{V}_m$. The second property expresses the *completeness* of the nested subspaces which fill the whole $\mathbf{L}^2(\mathbb{R})$ space. The third property indicates that the subspaces are not too redundant since their intersection only contain the zero element. The fourth property indicates *self-similarity in scale*, which means that all subspaces $\mathbf{V}_m$ are time-scaled versions of each other by a scaling factor of $\frac{1}{2}$. The fifth property demands *self-similarity in time*, which means that each subspace $\mathbf{V}_m$ is invariant under shifts by integer multiples of l. Finally, the sixth property is related to the existence of an orthonormal basis with integer shifts for each subspace.

Within the context of DWT, the approximation of a function $f(x)$ at a resolution 2^{-m} is defined as the orthogonal projection over the space $\mathbf{V}_m \in \mathbf{L}^2(\mathbb{R})$ with an expansion in a scaling orthogonal basis of the form:

$$f_{V_m}(x) = \sum_k a_{m,k}\phi_{m,k}(x) \qquad \forall k \in \mathbb{Z} \tag{2.1}$$

where $f_{V_m}(x)$ is the approximation of $f(x)$ at resolution 2^{-m}.

Note that in Equation (2.1) the orthonormal basis of the space $\mathbf{V}_m$ is constructed by dilating and translating a single function ϕ called the *scaling function* and whose translated and dilated versions are expressed by $\phi_{m,k}(x) = 2^{-m/2}\phi(2^{-m}x - k)$.

In Equation (2.1) the scaling function coefficients $a_{m,k}$ are expressed by:

$$a_{m,k} = \langle f(x), \phi_{m,k}(x) \rangle = \int_{-\infty}^{\infty} f(x)\phi_{m,k}(x)dx \qquad \forall m, k \in \mathbb{Z} \tag{2.2}$$

where the operator $\langle . \rangle$ is used to denote inner product in $\mathbf{L}^2(\mathbb{R})$.

In addition to the sequence $\{\mathbf{V}_m\}_{m\in\mathbb{Z}}$, DWT also considers spaces $\mathbf{W}_m \in \mathbf{L}^2(\mathbb{R})$ related to the orthogonal complement of $\mathbf{V}_m$ in $\mathbf{V}_{m+1}$ which can be formally expressed as:

$$\mathbf{V}_{m-1} = \mathbf{V}_m \oplus \mathbf{W}_m \tag{2.3}$$

The orthonormal basis of the space $\mathbf{W}_m$ is, similarly to the basis of $\mathbf{V}_m$, constructed by dilating and translating a single function ψ called the *wavelet function* and whose translated and dilated versions are expressed by $\psi_{m,k}(x) = 2^{-m/2}\psi(2^{-m}x - k)$. In this way, the orthogonal projection of $f(x)$ in the space $\mathbf{W}_m$ is obtained with a partial expansion in its wavelet basis:

$$f_{W_m}(x) = \sum_k d_{m,k}\psi_{m,k}(x) \qquad \forall k \in \mathbb{Z} \tag{2.4}$$

where the wavelet coefficients $d_{m,k}$ in Equation (2.4) are obtained by

$$d_{m,k} = \langle f(x), \psi_{m,k}(x) \rangle = \int_{-\infty}^{\infty} f(x)\psi_{m,k}(x)dx \qquad \forall m, k \in \mathbb{Z} \tag{2.5}$$

According to Equation (2.3) and Equation (2.4) the orthogonal projection of $f(x)$ on $\mathbf{V}_{m-1}$ can be decomposed as the sum of the orthogonal projections on $\mathbf{V}_m$ and $\mathbf{W}_m$.

$$f_{V_{m-1}}(x) = f_{V_m}(x) + f_{W_m}(x) \tag{2.6}$$

Note that in the DWT framework the spaces $\mathbf{V}_m$ and $\mathbf{W}_m$ are called *approximation* and *detail* spaces, respectively. Note also that for any resolution 2^{-m}, $\{\psi_{m,k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of $\mathbf{W}_m$ while for all resolutions, $\{\psi_{m,k}\}_{m,k\in\mathbb{Z}}$ is an orthonormal basis of $\mathbf{L}^2(\mathbb{R})$.

Specifically, DWT is based on the successive decomposition of each approximation $f_{V_m}(x)$ into a coarser approximation $f_{V_m+1}(x)$ plus the projection in the detail space $f_{W_m+1}(x)$, which formally can be expressed as:

$$f_{V_m}(x) = f_{V_{m+M}}(x) + \sum_{p=m+1}^{m+M} f_{W_p}(x) \tag{2.7}$$

where $2^{-(m+M)}$ is the resolution of the coarsest approximation of the analysis.

In the DWT framework, the function $f(x)$ is assumed to be at resolution 2^0, and in that sense, according to Equation (2.7), it can be expressed as the sum of a coarser approximation and the set of successive details:

$$f(x) = \sum_k c_{M,k}\phi_{M,k}(x) + \sum_{m=1}^{M}\sum_k d_{m,k}\psi_{m,k}(x) \tag{2.8}$$

In practice, DWT is usually implemented in a computationally efficient manner using a set quadrature mirror filters (Mallat's cascade algorithm) [59] whose basic idea is the representation of wavelet basis as a set of high-pass and low-pass filters in a filter bank structure [61].

2.6 Nonparametric Statistics

Since one of the main objectives of this thesis is to design domain independent time series and data stream representation algorithms, concepts from nonparametric statistics, which include all those techniques that do not rely on any particular distribution of the data, become the essential elements when constructing/designing possible algorithmic solutions.

It is worth noting that all the proposed algorithms rely on nonparametric statistical concepts. All these algorithms consider nonparametric density estimation as a fundamental building block. In this section the density estimation techniques relevant to the work proposed in this thesis are briefly reviewed.

2.6.1 Density Estimation

Density estimation is the problem of estimating a probability density function from some given observed data. This is a well studied problem for which several solutions have been presented in the literature. Density estimation techniques can be categorised into two main groups: parametric and nonparametric approaches. Within the nonparametric group, Kernel Density Estimators (KDE) [62], Finite Gaussian Mixtures (FGM) [63], Orthogonal Series Estimators (OSE) [64] and histograms [65] have been the focus of attention of the majority of the research community.

In this thesis the attention is focused on FGMs and the so called Wavelet Density Estimator (WDE) which is a special type of OSE.

2.6.1.1 Finite Gaussian Mixtures

Finite Gaussian mixtures-based density estimation techniques assume that the component distributions belong to the parametric family of Gaussian functions. They are expressed by:

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{M} \hat{w}_j \varphi_j(\mathbf{x}) = \sum_{j=1}^{M} \hat{w}_j \mathcal{N}(\mathbf{x}; \hat{\mu}_j, \hat{\Sigma}_j) \tag{2.9}$$

where M is the number of components in the mixture, $\mathcal{N}(\mathbf{x}; \hat{\mu}_j, \hat{\Sigma}_j)$ is a normal distribution with mean $\hat{\mu}_j$, covariance matrix $\hat{\Sigma}_j$ and $\hat{w}_j$ is the mixture weight, with the constraint $\sum \hat{w}_j = 1$.

The problem of estimating parameters in a finite Gaussian mixture has been extensively studied in the literature. This work considers the well-known EM algorithm. For further details the reader is referred to the seminal papers [40] in which the EM algorithm was formalised and to [66] in which it was applied to Gaussian mixtures, respectively.

2.6.2 Wavelet-based Density Estimation

Wavelet Density Estimators (WDEs) fall into the class of OSE methodology, originally introduced by Cêncov in [67]. In the context of OSE, an unknown square integrable density function $f(x)$ can be expressed as a convergent series of orthogonal basis functions:

$$f(x) = \sum_{j} b_j \varphi_j(x) \tag{2.10}$$

where $\{\varphi_j\}_{j \in \mathcal{J}}$ is a complete orthonormal system of basis functions for $\mathbf{L}^2(\mathbb{R})$, b_j is the coefficient of the j^{th} basis function and $\mathcal{J}$ is an appropriate set of indices that belongs to $\mathbb{Z}$. Then, if $X_1, X_2, ..., X_n$ are the realisations of a random variable X, then the coefficient b_j can be expressed as the expectation: $b_j = \langle f, b_j \rangle = \int \varphi_j(x) f(x) dx = E[\varphi_j(X)]$ with $j \in \mathcal{J}$. Consequently, the j^{th} series coefficient in an orthogonal series estimator can be approximated by

$$\hat{b}_j = \frac{1}{n} \sum_{i=0}^{n} \varphi_j(X_i) \tag{2.11}$$

and the corresponding approximated density can be expressed as:

$$\hat{f}(x) = \sum_{j} \hat{b}_j \psi_j(x) \tag{2.12}$$

Wavelet density estimators follow the concepts described in Equation (2.11) and Equation (2.12); however, within the wavelet framework the density can be represented as an orthogonal series of either scaling and wavelet functions or only scaling functions.

For the first alternative the corresponding density estimate can be formally expressed as:

$$\hat{f}(x) = \sum_{k} \hat{c}_{j_0,k} \phi_{j_0,k}(x) + \sum_{j=j_0}^{j_0+J} \sum_{k} \hat{d}_{j,k} \psi_{j,k}(x) \tag{2.13}$$

where $\{\phi_{j_0,k}(x)\}_{j_0 \in Z, k \in \mathcal{K}}$ plus $\{\psi_{j,k}(x)\}_{j \in Z, k \in \mathcal{K}}$ is a complete orthonormal basis system for $\mathbf{L}^2(\mathbb{R})$, with $\mathcal{K}$ denoting an appropriate set of indices that belongs to $\mathbb{Z}$, with $j_0 \in \mathbb{Z}$ referring to the index associated with the coarsest resolution of analysis 2^{-j_0}, and with $J \in \mathbb{Z}$ defining the number of decomposition levels. Since the estimator of Equation

(2.13) is based on the concept of multiresolution approximation (as introduced in Section 2.5), the scaling and wavelet functions have the form $\phi_{j_0,k}(x) = 2^{-j_0/2}\phi(2^{-j_0}x - k)$, and $\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k)$, respectively.

In Equation (2.13) $\hat{c}_{j_0,k}$ and $\hat{d}_{j,k}$, refer to the approximated or *empirical* coefficients for the scaling and wavelet functions, $\phi_{j_0,k}(x)$ and $\psi_{j_0,k}(x)$, respectively. If $X_1, X_2, ..., X_n$ are the realisations of a random variable X with an unknown square integrable density $f(x)$, then coefficients $\hat{c}_{j_0,k}$ and $\hat{d}_{j,k}$ can be approximated by:

$$\hat{c}_{j_0,k} = \frac{1}{n}\sum_{i=0}^{n} \phi_{j_0,k}(X_i) \tag{2.14}$$

$$\hat{d}_{j,k} = \frac{1}{n}\sum_{i=0}^{n} \psi_{j,k}(X_i) \tag{2.15}$$

Regarding the second alternative for the density, the one that only considers scaling functions, can be expressed as:

$$\hat{f}(x) = \sum_{k} \hat{c}_{j_0,k}\phi_{j_0,k}(x) \tag{2.16}$$

where the set $\{\phi_{j_0,k}(x)\}_{j_0 \in Z, k \in \mathcal{K}}$ alone is a complete orthonormal basis system for $\mathbf{L}^2(\mathbb{R})$, and j_0 is the index associated with the *base resolution* for the analysis.

In this thesis the estimator defined by Equation (2.16), which is the simplest linear WDE is the key block for the frameworks and algorithms proposed in Chapter 3 and Chapter 5. The reason for this, is that the reduced computation complexity of the estimator allows the construction of complex algorithms that involves multiple density estimation blocks.

The rate of convergence of orthogonal series estimators has been shown to be asymptotically optimal (see [68] for further details). Additionally, since wavelet basis functions possess good localisation properties, wavelet-based orthogonal series also have improved local approximation capabilities. Moreover, regarding the precision of this type of estimator, it depends on three aspects, the shape of the density to be estimated, the number of data items considered for the estimation and the number of the decomposition levels in

Equation (2.13) for the first alternative of the density, or the base resolution in Equation (2.16) for the second alternative of the density.

In the relevant literature, different variants of the general estimator described by Equation (2.13) to Equation (2.15) can be found (see, for instance [69–72]). These variants mainly differ in two basic aspects: the first one is the strategy they follow to select which terms in the series expansion should be kept, and the second one, is the way the series coefficients are thresholded. By applying these strategies, the construction of linear and nonlinear estimators is possible. A linear estimator can be constructed by keeping wavelet coefficients untouched. A non linear estimator can be constructed with soft or hard coefficients thresholding strategies. Note that that the density estimation approaches of Equation (2.13) to Equation (2.16) are based on a fixed-static data model and consequently they address the density estimation problem from a batch processing perspective. Furthermore, since wavelets are not a positive δ-sequence, density estimates may take negative values in regions where the sample is sparse. Different solutions for this problem have been reported in the literature, for example, taking the square root of the density [73] and using non-negative wavelets. Further and in depth discussion on these procedures can be found in [74].

Practical Implementation

The practical implementation of the estimators of Equation (2.13) and Equation (2.16) makes use of two well known algorithms: a) the recursive algorithm introduced by [75], and b) the Mallat's cascade algorithm [59]. The Daubechies-Lagarias algorithm (see Appendix A), is a numerical method for the calculation of scaling and wavelet values at a given point with a predefined precision. This algorithm is necessary since, for most of the compactly supported wavelet families, both scaling functions ϕ and wavelet functions ψ have no explicit or closed form representation. On the other hand, Mallat's cascade algorithm is a fast and optimised procedure for the implementation of the DWT based on the use of filter banks and nested spaces. For more details about these two procedures the reader is referred to the original sources as well as to [74].

In broad terms, the implementation of the wavelet-based estimator is performed by choosing in Equation (2.13) the index related to the coarsest resolution 2^{-j_0} and the number of decomposition levels J, then the scaling coefficients $\hat{c}_{j_0,k}$ and the wavelet coefficients $\hat{d}_{j,k}$

for $j \in \{j_0, j_0 + 1, \ldots, j_0 + J\}$ can be obtained by the use of Mallat's cascade algorithm starting from the high resolution scaling coefficients $\hat{c}_{J,k}$ firstly obtained by applying the Daubechies-Lagarias algorithm over the original data.

2.7 Final Remarks

In this chapter an overview of the theoretical foundations for the frameworks proposed in this thesis was presented. To this end, the different paradigms for pattern recognition and machine learning as well as the main concepts from KDD and data mining were reviewed. In addition to this, some algorithms from sparse signal processing and nonparametric statistics which are relevant to the work presented in this thesis were also discussed.

In the following chapter the general formulation for the proposed SGDs representation framework, as well as its suggested offline and online implementations, will be developed.

Representation precedes learning.

Leslie Valiant,
ACM Alan Turing Award winner 2010

Chapter 3

Structural Generative Descriptions for Time Series Classification

In this chapter a novel domain independent time series representation framework is presented. By combining statistical and structural pattern recognition paradigms, this framework describes the structure of time series in terms of different resolution probability-domain subpatterns. The proposed framework, referred to as Structural Generative Descriptions (SGDs), allows the subsequent supervised and unsupervised classification of the obtained time series representations using statistical decision theoretic classification and clustering methods. Two offline and two online algorithmic instantiations are also presented.

This chapter is organised as follows. Section 3.1 presents an introduction to the problem of time series classification as well as some preliminary aspects of the proposed SGDs framework. In Section 3.2 relevant work in the area of time series supervised/unsupervised classification is briefly reviewed. In Section 3.3 the proposed SGDs framework is presented. The corresponding off-line and online algorithmic instantiations are formulated in Section 3.4 and Section 3.5, respectively. Final remarks can be found in Section 3.6.

3.1 Introduction

Time series classification constitutes an important subset of data mining applications since a large number of domains involves this particular class of temporal data. Notable application domains are among others, medical signal analysis, speech recognition, fault condition monitoring, mining in temporal databases and robot sensor analysis.

Machine learning and data mining communities have extensively studied the problem of time series classification, resulting in a plethora of solutions and algorithms. However, the bulk of available statistical clustering and classification techniques have been formulated in the context of *static data* (data whose feature values do not change over time) [2]. In order to apply classification methods developed for static data to the context of time series, usually one of the following strategies is followed [2]. The first strategy is an algorithmic-based solution, in which the idea is to modify classification algorithms developed in the context of static data to allow them to handle data whose features change with time, this is generally accomplished by replacing the distance measure used for static data with one suitable for time series. The second strategy is a data-based solution relying on the conversion of time series data into static one and then directly apply classification techniques developed for static data. Note that, the high amount of data dependency present in time series data is generally ignored by readily available static data algorithms.

In this chapter, the attention is placed on investigating the impact of the data representation stage in the solution of the generic time series classification problem. To this end, a novel time series representation strategy that captures the inherent data dependency of time series is formulated. Representations based on this strategy can be easily incorporated into existing statistical classification algorithms. Specifically, by moving the structural time series representation to the probability domain, the proposed framework is able to combine statistical and structural pattern recognition paradigms in a novel fashion.

The proposed Structural Generative Descriptions (SGDs) framework has its foundations on the observation that in complex pattern recognition problems, in which the structural dependency is important, an effective strategy would be to describe each pattern in terms of simpler subpatterns and the relations among them [2]. The proposed representation, first decomposes time series patterns into simpler subpatterns, and then learns a probabilistic model for each of these subpatterns (note that each probabilistic model can also be divided into simpler elements or primitives). The representation is completed by defining the set of attributes and relations between primitives. In this context, the probabilistic models are named as probability domain generative subpatterns and, their corresponding simpler elements (primitives) are termed as probability domain generative primitives. The representation is hence providing a description on how input time series patterns are constructed from their given probabilistic subpatterns and/or primitives. Note also that fixed-length feature vectors are used to describe the probability domain subpatterns and primitives. This enables the applications of any of the well established statistical or decision theoretic techniques in a subsequent supervised/unsupervised classification stage.

The SGDs framework described here treats time series sequences as stochastic processes for which the probability density function contains all the statistical information required for its characterisation. If this is the case, time series patterns and subpatterns can be effectively characterised by considering their associated specific stochastic properties. Note that the assumption underlying the proposed approach is general and does not depend on the particular class of objects to be recognised. It holds for a great variety of time series-based applications for which the hypothesis of grouping patterns according to their corresponding structural generative properties is valid. Furthermore, the development of a SGDs framework is also motivated by the fact that structural and statistical pattern recognition frameworks possess complementary properties and then a combined approach overcome some of their associated deficiencies while exploiting some of their advantages.

The time series description strategy using the SGDs framework proposed in this chapter has three main advantages: 1) it provides a compact representation of time series patterns, 2) it allows the construction of domain-independent time series classification systems and, 3) it also provides a description of the generation process of input time series data.

In this chapter the SGDs framework is also extended to the online context, making it suitable for data stream applications. Specifically, two different online algorithms based on SGDs concepts are presented. Both algorithms comprise the same blocks as offline SGDs, that is, multiresolution decomposition and density estimation, but in addition, they include fast online formulations.

These algorithms only combine statistical and structural pattern recognition paradigms, but they also fulfil some of the basic requirements for data stream mining algorithms: 1) they process data in a fast and incremental way; 2) the updating time and the amount of memory needed are both constant, 3) they provide at any point in time a compact representation of each data stream, and 4) they allow both concept shift and concept drift detection.

3.2 Related Work

There is a great body of research, within machine learning and data mining communities, dedicated to investigate outstanding issues on supervised and unsupervised time series classification. However, none of the existing approaches has previously considered time series representations similar to the proposed SGDs.

In this section, the literature on time series supervised and unsupervised classification is reviewed using as reference the chosen time series representation. An experimental comparison of time series representation methods and distance measures can be found in [76], while an overview of time series clustering techniques can be found in [2].

3.2.1 Time Series Representations Found in Classification Approaches

According to the pattern recognition framework the time series representations used by classification approaches can be categorised into *statistical* or *structural*. The great majority of the representation approaches available fall within the framework of statistical pattern recognition which, by combining the categorisations introduced by [77] and [2], can be grouped into two categories. The first group are *descriptive* techniques which are based on the direct comparison of observations (*raw data-based* approaches) or the conversion of time series data into a reduced fixed-length feature vector (*feature-based* approaches), and the subsequent application of distance measures. The second category includes *inferential* techniques (also known as *model-based* approaches in [2]) that rely on the construction of statistical models for time series data and the posterior evaluation of dissimilarity measures, but in this case, with respect to the underlying generation process of time series patterns, generally assumed to be linear and Gaussian.

Within *raw data-based descriptive* approaches, techniques based on sampling [78], piecewise approximation [79] and salient point [80] can be distinguished. *Feature-based descriptive* methods can be categorised according to the transformation technique they employ or the domain in which the distance similarity measure is applied namely, autocorrelation, cross-correlation, Fourier, Wavelets, Principal Component Analysis (PCA) and Single Value Decomposition (SVD). On the other hand, *inferential* time series classification approaches can be grouped according to the statistical model in which they are based on, e.g. ARIMA, ARMA and Hidden Markov Models (HMMs).

Although, the majority of existing approaches fall within the framework of statistical pattern recognition, some domain-specific solutions can be found relying on structural pattern recognition concepts[1]. Relevant approaches within this category are the codebook of key sequences proposed in [82], the Trend Description Language [83], the waveform parsing system [84], as well as the Automatic Least Squares Error Signal Decomposition Algorithm [85] and the fuzzy structural pattern recognition system [86]. Note that the so called Symbolic Aggregate Approximation (SAX)[87], its novel versions the iSAX [88] and the iSAX+[89], as well as its symbolic variant reported in [90] fall also within this category. Shapelet transform-based techniques [91] which use shapelets as pattern primitives can be also considered structural approaches.

Here it is important to highlight that the proposed SGDs framework is based on a time series representation approach that is radically different from the one followed by the above reviewed structural techniques. In the SGDs framework subpatterns and primitives are extracted and analysed in the probability domain which makes easier to codify structural aspects of the time series in a statistical form. Note that traditional time series representations like, for example, the aforementioned SAX [87], its variants [88–90], and the shapelet techniques [91], are all of them based on extracting subpatterns in time domain.

3.2.2 SGDs vs Wavelet-domain Gaussian Mixtures Models

In speech recognition and sound processing a common feature extraction approach useful for data mining tasks is the so-called Mel-scale Frequency Cepstrum (MFC), which is a representation of the short-term power spectrum of the signal [92]. The MFC is based on the cosine transformation of the log power spectrum on a nonlinear scale of frequency. Note here that techniques based on the MFC approach, such as the popular wavelet domain

[1] Since *structural* techniques are based on the conversion of time series data into symbolic form, they are also referred in the literature as *symbolic* techniques [81]

Gaussian Mixture Models of Mel-scale Frequency Cepstral Coefficients (wavelet-MFCCs-GMM) [93–95], may look similar to the proposed SGDs framework since both combine a wavelet decomposition block with a Gaussian Mixture Model (GMM). However, there are two fundamental differences between the two techniques. The first difference is that different features are used for each technique, in MFCCs-GMM techniques they are the MFC coefficients whereas in the proposed algorithm the vector of features includes the parameters of each of the Gaussian Models employed to describe each pattern. Note also that Gaussian models play a different role in these approaches, in the proposed approach they are used for feature extraction whereas in techniques based on MFCCs-GMM they are employed as a part of the classifier. The second difference is that, for the classifier, a different type of learning is employed. Techniques based on MFCCs-GMM rely on a classifier with generative learning. In contrast, discriminative learning is used in the proposed SGDs approach. The third difference is that, although techniques based on MFCCs-GMM consider some structural ideas (decomposition input patterns into subpatterns) they are not fully formulated using structural concepts such as patterns, subpatterns, primitives, primitive's attributes, primitive's relations, as in the proposed approach.

3.2.3 Data Streams

With the expansion of the Internet and the progress of hardware technology a new class of temporal data objects, referred to as data streams, is becoming pervasively and increasingly important in a wide variety of applications, notable examples are: sensor networks, network monitoring, transaction log analysis, financial tickers and web usage. What is common in all these applications is that data arrive in a continuous online time-varying fashion at a rapid rate and it is not feasible to exchange or to store all the arriving data in traditional database management systems (DBMS) to operate on it [96].

In a formal way, a *data stream* is a real-time, continuous, ordered and unbounded sequence of items whose order is implicit when it is represented by the arrival time or explicit when it is indicated by time stamps [97]. This new class of data leads to three particular computational challenges [98]. The first challenge is that algorithms designed to work with this type of data should process each data item only once, that is, they should work with only one pass of the data. The second challenge is related to the fact that algorithms should have fixed space and time computational complexity, independent of the amount of data to be processed and independent of the time horizon for the analysis. Finally, the third challenge is associated with the temporal component of the data, here algorithms should take into account time dependent relations as well as the evolution of the underlying data.

Traditional data mining techniques are designed to work with static records which generally have no predefined notion of time [2], their formulation is based on the storage and analysis of fixed static data archives that allow complex mining operations based on multiple passes of the data. However, for the case of data streams, the computational challenges inherent to this particular class of data, prevent the direct application of traditional data mining methods to this domain.

There is a large volume of literature which has been dedicated to design data mining methods suitable for streaming data (see for instance [98, 99] for some reviews). However, most of the work has been focused on designing, adapting or improving the so called data mining task algorithm, rather than constructing more robust data representations that serve as input for such algorithms. In general, in previously proposed algorithms, raw data is converted into characteristic or representative numeric features using incremental feature extraction, selection or reduction blocks and then mining algorithms are applied over these features. Usually, the resulting features do not take into account any temporal/structural relation or dependency present in the data.

3.2.4 Data Stream Mining Approaches

Different data stream mining problems have been thoroughly studied in the literature, resulting in a plethora of models, algorithms and applications. Different taxonomies can be used to categorise the voluminous work within the data stream mining framework. Aggarwal [98] and Gaber et al. [100] categorise available methods according to the data mining problem they address, that is, clustering [101–106], classification [107, 108], indexing [109, 110], frequent pattern mining [111–115], change detection [9, 116–119], summarisation (including sketching, load shedding and synopsis construction) [120–130], and forecasting [131].

Additionally, Gaber et al. [100] groups techniques according to their theoretical foundations, distinguishing between data-based techniques, which focus on the summarisation of streaming data, and task-based techniques which put the attention on the design of algorithms suitable for data streams. Within data-based techniques there are methods based on sampling [132], load shedding [133, 134], sketching [123, 124, 135], synopsis construction [136, 137], aggregation [106, 107], whereas within the task-based category there are approximation algorithms, sliding window-based approaches [138, 139] and algorithm output granularity [140, 141].

Since in this thesis the data streams mining problem is approached from the pattern recognition perspective, the attention is placed on the way data streams are represented prior to the application of the data mining task algorithm. For this reason, in this section, previous work in the field is reviewed and grouped according to the representation used for the algorithms following underlying concepts from pattern recognition and time series data mining frameworks [2, 77]. Note that, since it is outside the scope of this chapter to review all the existing work in the area, in this section, only relevant references to the proposed SGDs algorithms are selected.

In general, from the point of view of the descriptors employed, two main families of data stream mining algorithms can be distinguished in the literature: *statistical* approaches and *structural* techniques. Within the statistical category, representations in turn can be grouped into *descriptive* and *inferential* subcategories. On the one hand, *statistical descriptive* techniques, which are related to the data-based techniques of [100], are based on the extraction of a reduced number features or the summarisation of streaming data, and the subsequent application of distance measures among the features. *Statistical inferential* techniques, on the other hand, rely on the construction of statistical models for incoming streaming data and the evaluation of dissimilarity measures with respect to the underlying generation process of data streams patterns.

Within *statistical descriptive* approaches, there are techniques working directly with raw streaming data, that is, sampling [132], load shedding [133, 134], aggregation [106, 107], and methods based on transforming data streams to a different domain such as autocorrelation [138], Wavelets [142, 143], Haar Wavelet coefficient [102], DFT [103, 144], variance [145]. *Statistical inferential* data stream mining approaches can be grouped according to the statistical model in which they are based on, that is, Pearson's correlation coefficient [101], mean standard deviation and correlation [138], as well as local correlation integral

[146]. Note that hybrid approaches combining statistical inferential and statistical descriptive ideas can also be found in the literature. Examples of such approaches are: AR-DWT [147], the regression-based method reported in [102], the histogram-based techniques of [148, 149], as well as the autoregressive model proposed in [150].

Although, the majority of existing approaches fall within the framework of statistical pattern recognition, some domain-specific solutions can be found incorporating some structural pattern recognition concepts. Relevant approaches within this category are: structure aware sampling [151], and the small-space algorithm introduced in [105].

In the literature techniques including some multiresolution concepts can also be distinguished. Even though they are not explicitly defined using structural terms, they still incorporate some structural ideas. One of the most relevant techniques within this category is AWSOM [147] which is based on DWT concepts.

It is important to note that the computational restrictions inherent to data streams processing algorithms prevent the application of complex structural representations for this type of data. This is the reason why most of the work regarding data stream representation is based on simple statistical features such as subsampling [132], statistical measures [145] or correlation coefficients [101], all of them with a reduced computational burden. Note that the few structural-based representation approaches available for data streams are also constrained by the computational factor, and hence the subpatterns and primitives extraction is carried out, in all these approaches, in the time domain. In contrast, the proposed SGDs-based data stream representations, which are based on extracting subpatterns and primitives in the probability domain, manage to combine robust structural and statistical aspects without compromising computational complexity.

3.3 The Structural Generative Descriptions (SGDs) Framework

The proposed SGDs times series representations comprise two main stages, namely: multiresolution decomposition and density estimation. In the multiresolution decomposition stage input time series are decomposed into subpatterns at different resolutions using a given decomposition transform. In the density estimation stage, the obtained subpatterns are mapped into the probability domain by using a selected density estimation technique.

A key point of the time series description method proposed here, is the extraction of a representation of time series data based on a combined time-domain and probability-domain structural procedure in which the pattern decomposition is done in the time domain while pattern analysis and primitives extraction are performed in the probability domain. This procedure is depicted in Figure 3.1. Note that since in the proposed SGDs representation framework primitives of probability subpatterns are assumed to be the *base* functions (for example, Gaussian functions or orthogonal basis functions) used by the selected density estimation technique, then finding primitive's attributes and primitive's relations can be done by means of semi-parametric and nonparametric density estimation techniques. Note also that although the proposed SGDs representations are not formulated in linguistic terms, they are structural in essence. This remark is in accordance to the findings reported in [152], where the term structural pattern recognition is meant to refer to all methodologies which attempt to describe objects in terms of their parts and the juxtaposition relations between them.

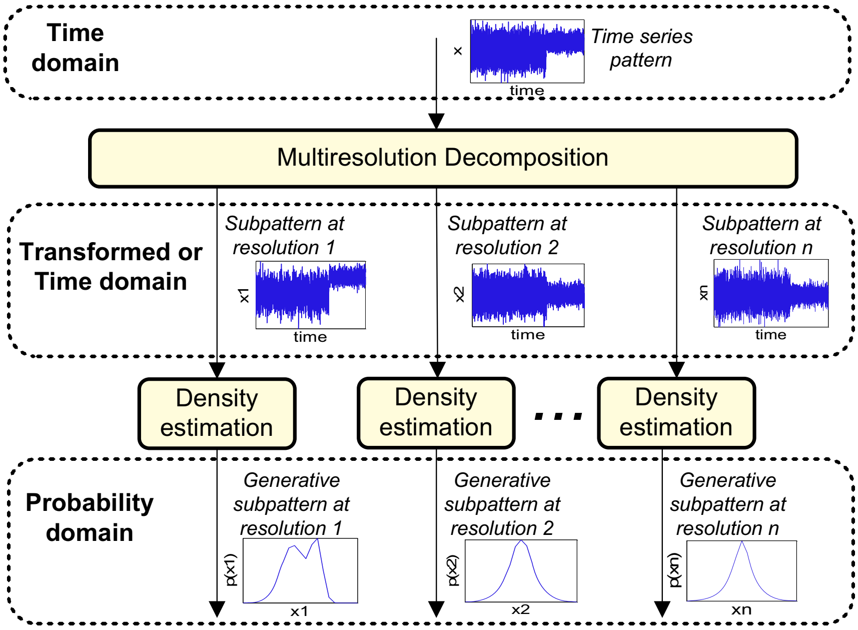

Time domain
x
time
Time series pattern
Multiresolution Decomposition
Transformed or Time domain
Subpattern at resolution 1
x1
time
Subpattern at resolution 2
x2
time
Subpattern at resolution n
xn
time
Density estimation
Density estimation
. . .
Density estimation
Probability domain
Generative subpattern at resolution 1
p(x1)
x1
Generative subpattern at resolution 2
p(x2)
x2
Generative subpattern at resolution n
p(xn)
xn

FIGURE 3.1: Proposed SGDs for time series.

3.3.1 The Structural Generative Description Block

The proposed SGDs assume a set of N univariate time series $\mathbf{X} = \{\mathbf{x}^i\}, i \in \mathcal{N} = \{1, 2, \ldots, N\}$, wherein each $\mathbf{x}^i = \{x^i(t)\}$ is an ordered sequence of n real valued observations taken at discrete times $t \in \mathcal{T} = \{1, 2, \ldots, n\}$. The objective of the description task is to extract, for each time series $\mathbf{x}^i$, a fixed-length feature vector $\mathbf{f}_{x^i}$ suitable to perform the subsequent supervised or unsupervised classification tasks.

The first stage of the SGDs framework is a multiresolution transformation Γ^* that decomposes the input time series pattern $\mathbf{x}^i = \{x^i(t)\}_{t\in\mathcal{T}}$ into a finite set $\{\mathbf{x}^i_p\}_{p\in\mathcal{P}}$ of P different time-domain resolution versions of the input pattern. For this, consider that initially $\mathbf{x}^i$ is decomposed according to the following general equation:

$$\mathbf{x}^i_m = \{x^i_m(t)\} = \Gamma(\mathbf{x}^i; m) \tag{3.1}$$

where the index $m \in \mathcal{M} = \{M, M-1, \dots, 1\}$ with $\mathcal{M} \subset \mathbb{Z}$, is associated with the resolution and M denotes the coarsest resolution of the decomposition process. If, in order to avoid redundancy, instead of working directly with subpatterns $\mathbf{x}^i_m$ the corresponding differences $\tilde{\mathbf{x}}^i_m$ between consecutive $\mathbf{x}^i_m$ are considered:

$$\tilde{\mathbf{x}}^i_m = \mathbf{x}^i_{m-1} - \mathbf{x}^i_m \tag{3.2}$$

Then the set $\{\tilde{\mathbf{x}}^i_M, \tilde{\mathbf{x}}^i_{M-1} \dots, \tilde{\mathbf{x}}^i_1\}$ is the set of different resolution structural time-domain subpatterns containing complementary information for $\mathbf{x}^i_M$. This work considers a multiresolution transformation Γ based on the concept of nested subspaces with an approximation operator that follows the properties described in Chapter 2 for multiresolution approximations. Hence, the input pattern $\mathbf{x}^i$ can be assumed similar to the highest resolution pattern $\mathbf{x}^i_0$, and in that sense it can be alternatively expressed as:

$$\mathbf{x}^i = \mathbf{x}^i_0 = \mathbf{x}^i_M + \tilde{\mathbf{x}}^i_M + \tilde{\mathbf{x}}^i_{M-1} \dots + \tilde{\mathbf{x}}^i_1 \tag{3.3}$$

Note that Equation (3.3) shows the structural characteristics of the proposed multiresolution decomposition, in which the input pattern $\mathbf{x}^i$ is constructed by combining its corresponding subpatterns associated with different resolutions.

The second stage of the proposed SGDs is the mapping of subpatterns of Equation (3.3) into the probability domain by estimating their probability density functions. Let us consider the set of subpatterns defined by:

$$\{\bar{\mathbf{x}}_p^i\}_{p\in\mathcal{P}} = \{\mathbf{x}_M^i, \tilde{\mathbf{x}}_M^i, \tilde{\mathbf{x}}_{M-1}^i \cdots, \tilde{\mathbf{x}}_1^i\} \tag{3.4}$$

where $\{\bar{\mathbf{x}}_p^i\}_{p\in\mathcal{P}}$ with $\mathcal{P} = \{1, 2, \ldots, P\}$ and $P = M + 1$ denotes the set of P multiresolution time domain subpatterns of the time series pattern $\mathbf{x}^i$ containing complementary information at different resolutions, where $\bar{\mathbf{x}}_1^i = \mathbf{x}_M^i$ and $\bar{\mathbf{x}}_P^i = \tilde{\mathbf{x}}_1^i$, here $\bar{\mathbf{x}}_p^i$ refers to the p time domain subpattern of the input pattern $\mathbf{x}^i$.

Then the approximated probability density for the subpattern $\bar{\mathbf{x}}_p^i$ using the estimator parameters $\boldsymbol{\theta}_p^i = \{\theta_k^i\}_{k\in\mathcal{K}}$, with $\mathcal{K} = \{1, \ldots, K\}$ and K denoting the number of parameters, is expressed by $\hat{f}_p^i(\bar{\mathbf{x}}_p^i, \boldsymbol{\theta}_p^i)$. Note that a key assumption in the framework proposed here is that the estimated density $\hat{f}_p^i(\bar{\mathbf{x}}_p^i; \boldsymbol{\theta}_p^i)$ is a probability domain version of $\bar{\mathbf{x}}_p^i$, denoted as $\breve{\mathbf{x}}_p^i$, and in that sense, it is a probability domain subpattern for the time series pattern $\mathbf{x}^i$. Consequently, the set $\{\breve{\mathbf{x}}_p^i\}_{p\in\mathcal{P}}$ is the set of probability domain subpatterns of $\mathbf{x}^i$.

Note that in the proposed framework, there are no assumptions about the functional form of the probability densities employed, and as a consequence their estimation is not restricted to a particular parametric or nonparametric technique. The only requirement is a sparse density representation, which means that for the subpattern $\bar{\mathbf{x}}_p^i$ the estimated density $\hat{f}_p^i(\bar{\mathbf{x}}_p^i; \boldsymbol{\theta}_p^i)$ is expressed by a reduced number of parameters $\boldsymbol{\theta}_p^i$. Since probability density functions embody all the information for the characterisation of stochastic processes, the obtained probability domain subpatterns $\{\breve{\mathbf{x}}_p^i\}_{p\in\mathcal{P}}$ can be used to generate or synthesise time domain subpatterns with similar statistical properties as $\{\bar{\mathbf{x}}_p^i\}_{p\in\mathcal{P}}$. This property makes the proposed probability domain subpatterns essentially generative.

Although, there are different procedures for density estimation in the literature, the three most commonly used methods (which are kernel-based, Gaussian mixtures and orthogonal series) can be expressed as the weighted combination of k *base* functions. These base functions could be kernels, Gaussian functions or orthogonal functions, depending on the case. Considering this, the estimated density for the time domain subpattern $\bar{\mathbf{x}}_p^i$ can be expressed according to Equation (3.5).

$$\hat{f}_p^i(\bar{\mathbf{x}}_p^i; \boldsymbol{\theta}_p^i) = \sum_k \alpha_{p,k}^i h(\bar{\mathbf{x}}_p^i; \beta_{p,k}^i) \tag{3.5}$$

where $\alpha_{p,k}^i$ represents the weight for the k^{th} term, and $h(\bar{\mathbf{x}}_p^i; \beta_{p,k}^i)$ denotes the *base* function with parameters represented by $\beta_{p,k}^i$. Note, that in the left side of the equation the set of parameters $\boldsymbol{\theta}_p^i$ is equal to $\{\alpha_{p,k}^i, \beta_{p,k}^i\}_{k \in \mathcal{K}}$.

For the SGDs representations proposed here, the set of *base* functions $\{h(\bar{\mathbf{x}}_p^i; \beta_{p,k}^i)\}_{k \in \mathcal{K}}$ constitute *structural generative primitives* for the probability domain subpattern $\breve{\mathbf{x}}_p^i$ while the sets $\{\beta_{p,k}^i\}_{k \in \mathcal{K}}$ and $\{\alpha_{p,k}^i\}_{k \in \mathcal{K}}$, are the corresponding set of primitive's attributes and the set primitive's relations, respectively. While the former set specifies particular characteristics of the primitive, the latter set describes the way primitives are related in order to construct a given probability domain subpattern. In this work the set of probability domain subpatterns is constructed using the same primitive (e.g. kernels, Gaussian functions, orthogonal functions), but with different numerical attributes. The SGDs framework is based on describing the time series subpattern $\bar{\mathbf{x}}_p^i$ using its primitive attributes set $\{\beta_{p,k}^i\}_{k \in \mathcal{K}}$ and its relations set $\{\alpha_{p,k}^i\}_{k \in \mathcal{K}}$. In this way, the input time series pattern $\mathbf{x}^i$ is described using the set of attributes sets $\{\{\beta_{p,k}^i\}_{k \in \mathcal{K}}\}_{p \in \mathcal{P}}$ of structural probability domain primitives together with the corresponding set of relations sets $\{\{\alpha_{j,k}^i\}_{k \in \mathcal{K}}\}_{p \in \mathcal{P}}$, both grouped together in $\mathbf{f}_{x^i}$, which is a fixed-length feature vector.

3.3.2 Statistical Discriminative Classification using SGDs

The supervised/unsupervised classification of time series based on the proposed SGDs does not require a grammar or a parsing algorithm, since the descriptions provided are fixed-length pattern representations, and as a consequence, they allow the subsequent use of well established techniques from statistical decision theory. Hence, the supervised or unsupervised classification block can be formulated in general terms as the task of finding a discriminant function $g(\mathbf{f}_{x^i})$ which determines the class membership of the generative structural description of the pattern $\mathbf{x}^i$ expressed by the feature vector $\mathbf{f}_{x^i}$.

3.4 Two Offline Algorithmic Instantiations

In this section, two algorithms based on the proposed SGDs framework are developed. They differ in the density estimation technique selected for the structural generative descriptor. The first algorithm, called SGDG, relies on the use of Finite Gaussian Mixtures (FGM) which belongs to the semi-parametric category of density estimation techniques.

The second algorithm, referred to as SGDW, is founded on Wavelet Density Estimators (WDE) and belongs to the nonparametric density estimation category. Both algorithms use the Discrete Wavelet Transform (DWT) for the multiresolution decomposition stage.

Note that DWT has been selected since: 1) it is the most popular multiresolution decomposition method, 2) it has strong theoretical foundations, 3) there are fast algorithms available and 4) it is a non redundant wavelet transform. Regarding the WDE and FGM estimators they have been considered because: 1) among the sparse density estimators, these techniques are among the simplest to implement and 2) they consider different *base* functions namely, wavelets and Gaussian functions, and in this way they show how the proposed SGDs framework can be implemented using different generative primitives.

3.4.1 Wavelet-based Multiresolution Decomposition

The first stage in the proposed SGDs-based algorithms is the multiresolution decomposition of input time series patterns using DWT. For this, let the time series $\mathbf{x}^i$ be decomposed into scaling and wavelet coefficients according to DWT equations:

$$a^i_{M,l} = \langle x^i(t), \phi_{M,l}(t) \rangle \tag{3.6}$$

$$d^i_{m,l} = \langle x^i(t), \psi_{m,l}(t) \rangle \tag{3.7}$$

where $M \in \mathbb{Z}$, $m \in \mathcal{M} = \{M, M-1, \ldots, 1\}$, $l \in \mathcal{L} \subset \mathbb{Z}$ and the operator $\langle . \rangle$ denotes the inner product in $L^2(\mathbb{R})$. In Equation (3.6) and Equation (3.7), $a^i_{M,l}$ and $d^i_{m,l}$ are the corresponding scaling and wavelet coefficients associated with resolutions 2^{-M} and 2^{-m}, respectively. With the scaling function defined as $\phi_{M,l}(t) = 2^{-M/2}\phi(2^{-M}t - l)$ and the wavelet functions expressed by $\psi_{m,l}(t) = 2^{-m/2}\psi(2^{-m}t - l)$. Here the index M is related to the coarsest resolution 2^{-M}.

Therefore, the time domain subpattern corresponding to the coarsest resolution 2^{-M} is $\mathbf{x}^i_M = \{x^i_M(t)\}_{t \in \mathcal{T}}$ where $x^i_M(t)$ is calculated using:

$$x^i_M(t) = \sum_l a^i_{M,l}\phi_{M,l}(t) \tag{3.8}$$

Note that, within the DWT framework, detail coefficients $d^i_{m,l}$ and their corresponding reconstructed time domain subpatterns $\tilde{\mathbf{x}}^i_m = \{\tilde{x}^i_m(t)\}_{t\in\mathcal{T}}$ are the differences between subpatterns associated with consecutive resolutions $\tilde{\mathbf{x}}^i_m = \mathbf{x}^i_{m-1} - \mathbf{x}^i_m$. Therefore, $\tilde{x}^i_m(t)$ can be directly expressed as:

$$\tilde{x}^i_m(t) = \sum_l d^i_{m,l}\psi_{m,l}(t) \tag{3.9}$$

In the DWT context, the highest resolution of the analysis is 2^0 and its associated pattern $\mathbf{x}^i_0$ is assumed equal to the input pattern $\mathbf{x}^i$. Hence, it can be alternatively expressed as:

$$\mathbf{x}^i = \mathbf{x}^i_0 = \mathbf{x}^i_M + \tilde{\mathbf{x}}^i_M + \tilde{\mathbf{x}}^i_{M-1} \ldots + \tilde{\mathbf{x}}^i_1 \tag{3.10}$$

Equation Equation (3.10) shows the structural organisation of the multiresolution decomposition, in which the time series pattern $\mathbf{x}^i$ is constructed by combining its corresponding subpatterns at different resolutions.

Note that all the information of $\mathbf{x}^i_M$ and $\tilde{\mathbf{x}}^i_m$ is already contained in the corresponding set of scaling coefficients $\mathbf{a}^i_M = \{a^i_{M,l}\}_{l\in\mathcal{L}}$ and wavelet coefficients $\mathbf{d}^i_m = \{d^i_{m,l}\}_{l\in\mathcal{L}}$. In that sense $\mathbf{a}^i_{M,l}$ and $\mathbf{d}^i_{m,l}$ can be viewed as condensed representations of $\mathbf{x}^i_M$ and $\tilde{\mathbf{x}}^i_m$ and then considered wavelet domain subpatterns of $\mathbf{x}^i$. For clarity consider:

$$\{\bar{\mathbf{x}}^i_p\}_{p\in\mathcal{P}} = \{\mathbf{a}^i_M, \mathbf{d}^i_M, \mathbf{d}^i_{M-1} \ldots, \mathbf{d}^i_1\} \tag{3.11}$$

where $\{\bar{\mathbf{x}}^i_p\}_{p\in\mathcal{P}}$ with $\mathcal{P} = \{1, 2, \ldots, P\}$ with $P = M + 1$ denotes the set of P multiresolution wavelet domain subpatterns of the time series pattern $\mathbf{x}^i$ containing complementary information at different resolutions. Here $\bar{\mathbf{x}}^i_1 = \mathbf{a}^i_M$ and $\bar{\mathbf{x}}^i_P = \mathbf{d}^i_1$.

Summarising, the proposed time series representation starts with a multiresolution transformation Γ that decomposes the time series $\mathbf{x}^i$ into a finite set of wavelet-domain subpatterns at different resolutions $\{\bar{\mathbf{x}}^i_p\}_{p\in\mathcal{P}}$.

3.4.2 Density Estimation

The second stage in the SGDs framework is mapping the wavelet domain subpatterns of Equation (3.11) into the probability domain. This is done by estimating their corresponding probability densities using the WDE and FGM algorithms.

3.4.2.1 Wavelet-based Density Estimator (WDE)

The WDE algorithm relies on representing the probability density as an orthogonal series of scaling and wavelet functions. This thesis considers the WDE with the lowest computational complexity which only considers scaling functions ϕ. Note that more complex density estimators will significantly impact the computational burden of the proposed SGDs algorithm since it considers the density estimation of every subpattern in a multiresolution structure.

In the SGDs context, the probability domain subpattern $\breve{\mathbf{x}}_p^i$ is the density of the time domain subpattern $\bar{\mathbf{x}}_p^i = \{\bar{x}_p^i(v_p)\}_{v_p \in \mathcal{V}_p}$; $\mathcal{V}_p = \{1, \ldots, |\bar{\mathbf{x}}_p^i|\}$ evaluated at point u_q, where the symbol $|\bar{\mathbf{x}}_p^i|$ denotes the cardinality of $\bar{\mathbf{x}}_p^i$. The subpattern $\breve{\mathbf{x}}_p^i$ is then defined by the following WDE equation:

$$\breve{\mathbf{x}}_p^i = \hat{f}_p^i(u_q) = \sum_k \hat{c}_{p,j_0,k}^i \phi_{j_0,k}(u_q) \tag{3.12}$$

where $\phi_{j_0,k}(u_q) = 2^{-j_0/2}\phi(2^{-j_0}u_q - k)$ is the scaling function associated with the *base resolution* 2^{-j_0} with $j_0 \in \mathbb{Z}$ and $k \in \mathcal{K} \subset \mathbb{Z}$. Here $u_q \in \mathcal{U} = \{u_1, \ldots, u_Q\}$; with $U \subset \mathbb{R}$ is a set of Q points in which the corresponding density is evaluated. In Equation (3.12), the scaling function coefficients $\hat{c}_{p,j_0,k}^i$ are estimated according to

$$\hat{c}_{p,j_0,k}^i = \frac{1}{n} \sum_{v_p} \phi_{j_0,k}(\bar{x}_p^i(v_p)) \tag{3.13}$$

Note that for convenience WDEs are usually restricted to the space $L^2([0,1])$ and in that sense, the input data requires to be normalised to the interval [0,1]. In this way, at resolution 2^{-j_0}, the set of translation parameters is $k \in \mathcal{K} = \{-(2n_\phi - 1), \ldots, 2^{j_0}\} \subset \mathbb{Z}$, where n_ϕ denotes the order of the scaling function filter (for instance, $n_\phi = 1$ for $db1$, $n_\phi = 2$ for $db2$, and so on).

3.4.2.2 Finite Gaussian Mixtures (FGM) Estimator

The second density estimation method suggested is the FGM estimator which assumes that the component distributions belong to the parametric family of Gaussian functions. Here, the probability domain subpattern $\breve{\mathbf{x}}_p^i$ which is the probability density of the wavelet domain subpattern $\bar{\mathbf{x}}_p^i = \{\bar{x}_p^i(v_p)\}_{v_p \in \mathcal{V}_p}$ with $\mathcal{V}_p = \{1, \ldots, |\bar{\mathbf{x}}_p^i|\}$, can be expressed according to the following equation:

$$\begin{aligned} \breve{\mathbf{x}}_p^i = \hat{f}_p^i(u_q) \quad &= \textstyle\sum_{k=1}^{K} \hat{w}_{p,k}^i \varphi_{p,k}^i(u_q) \\ &= \textstyle\sum_{k=1}^{K} \hat{w}_{p,k}^i \mathcal{N}(u_q; \hat{\mu}_{p,k}^i, \hat{\Sigma}_{p,k}^i) \end{aligned} \tag{3.14}$$

where K is the number of components in the mixture, $\mathcal{N}(u_q; \hat{\mu}_{p,k}^i, \hat{\Sigma}_{p,k}^i)$ is a normal distribution with mean $\hat{\mu}_{p,k}^i$, covariance matrix $\hat{\Sigma}_{p,k}^i$ and mixture weight $\hat{w}_{p,k}^i$ evaluated at point $u_q \in \mathcal{U}$. Here the mixture weights have the constraint $\sum_{k=1}^{K} \hat{w}_{p,k}^i = 1$.

3.4.3 Remarks on the SGDs Algorithmic Instantiations

The block diagram for the proposed algorithms is presented in Figure 3.2. Note that since the SGDW algorithm relies on linear WDE as density estimate, primitives are the scaling functions $\phi(\cdot)$, while their parameters $\{k\}_{k \in \mathcal{K}}$ and their coefficients $\{\hat{c}_{p,j_0,k}^i\}_{k \in \mathcal{K}}$ are primitive's attributes and primitive's relations, respectively. In contrast, in the SGDG algorithm, primitives are the Gaussian functions $\varphi(.)$ employed by the FGM density estimator, with $\{\hat{\mu}_{p,k}^i\}_{k \in \mathcal{K}}$ and $\{\hat{\Sigma}_{p,k}^i\}_{k \in \mathcal{K}}$ denoting primitive's attributes and $\{\hat{w}_{p,k}^i\}_{k \in \mathcal{K}}$ accounting for primitive's relations.

3.4.4 Features and Normalisation Strategies

In this section different alternatives for the feature vector are proposed. They differ in the normalisation strategy they follow as well as in the characteristics of the primitive selected as features.

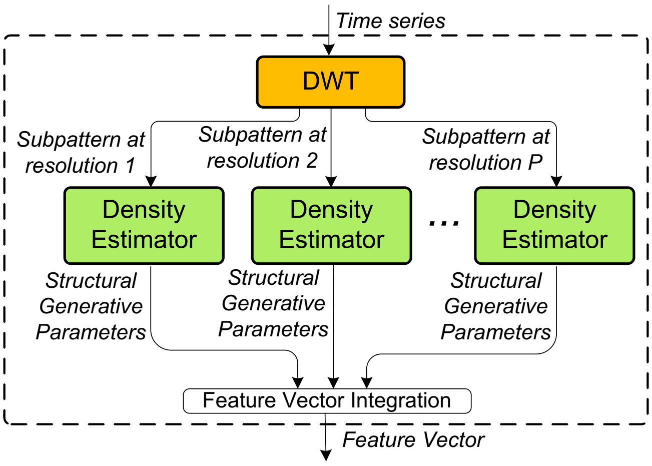

Time series
DWT
Subpattern at resolution 1
Subpattern at resolution 2
Subpattern at resolution P
Density Estimator
Density Estimator
...
Density Estimator
Structural Generative Parameters
Structural Generative Parameters
Structural Generative Parameters
Feature Vector Integration
Feature Vector

FIGURE 3.2: Proposed SGDW (Density Estimator=WDE) and SGDG (Density Estimator=FGM) algorithms.

Normalisation is particularly required for SGDW algorithm to restrict the evaluation of basis functions in the density estimation stage to the interval $[0, 1]$. Each data point $\{\bar{x}^i_p(v_p)\}_{v_p \in \mathcal{V}_p}$ of the p time domain subpattern $\mathbf{x}^i_p$ is normalised according to the equation $\hat{x}^i_p(v_p) = (\bar{x}^i_p(vp) - b^{lower}_p)/r_p$; where $\hat{x}^i_p(v_p)$ is a normalised data point and the interval r_p is defined by $r_p = b^{upper}_p - b^{lower}_p$. With b^{upper}_p and b^{lower}_p denoting the upper and the lower observation bounds, which are related to the smallest and the greatest observation that can be included in the WDE density estimate. Note that all the data points outside the interval r_p will be ignored by the WDE algorithm since they are outside the support of the corresponding basis functions.

In this chapter global and local normalisation strategies are proposed. These strategies basically differ in the selection of the upper and lower observation bounds, b_u and b_l. On the one hand, global normalisation considers $b^{lower}_p = \mu_{\mathbf{X}_p} - 3\sigma_{\mathbf{X}_p}$ and $b^{upper}_p = \mu_{\mathbf{X}_p} + 3\sigma_{\mathbf{X}_p}$, where $\mu_{\mathbf{X}_p}$ and $\sigma_{\mathbf{X}_p}$ denotes the mean and standard deviation of all the p time domain subpattern $\bar{\mathbf{x}}^i_p$ of all the time series in a given data set $\mathbf{X}$. This strategy, which is applicable in cases where the whole data set is available, enables the use of the same set of basis functions for the calculation of each WDE of all time series in a data set and in that sense, it has a reduced computational burden . On the other hand, in local normalisation $b^{lower}_p = \mu_{\mathbf{x}^i_p} - 3\sigma_{\mathbf{x}^i_p}$ and $b^{upper}_p = \mu_{\mathbf{x}^i_p} + 3\sigma_{\mathbf{x}^i_p}$, where $\mu_{\mathbf{x}^i_p}$ and $\sigma_{\mathbf{x}^i_p}$ are the mean and standard deviation of a particular time domain subpattern $\bar{\mathbf{x}}^i_p$. Here, since different bases functions are employed for each time series, additional parameters or attributes need to be included in the feature vector. As a result, this strategy is computationally more expensive than the global one.

Among the advantages and disadvantages of choosing a particular feature and normalisation strategy, it can be highlighted that working directly with wavelet and scaling coefficients (for the case of SGDW), or working with means, covariances and mixture weights (for the case of SGDG), is less computationally expensive than working with the reconstructed density function. Note here that the coefficients/parameters can be seen as a compact representation of the density. On the other hand, different normalisation strategies, such as for example, global and local, offer different discrimination capabilities, as it will become evident in the empirical evaluation of Chapter 4.

3.4.4.1 Features and Normalisation Strategies for SGDW

For the case of SGDW the following three strategies are studied:

Global coefficients as features: It considers that all probability domain subpatterns $\breve{\mathbf{x}}_p^i$ for all time series in a data set are constructed using a set of scaling functions $\{\phi_{j_0,k}\}_{k\in\mathcal{K}}$ with the same parameters k's. Then, by defining a vector of scaling coefficients $\mathbf{c}_p^i = \{\hat{c}_{p,j_0,k}^i\}_{k\in\mathcal{K}}$, the corresponding feature vector for pattern $\mathbf{x}^i$ is

$$\mathbf{f}_{x^i} = [\mathbf{c}_1^i \bullet \ldots \bullet \mathbf{c}_P^i] \tag{3.15}$$

where the symbol $\bullet$ denotes concatenation.

Local densities as features: The second alternative for the feature vector involves working directly with the probability domain subpattern $\breve{\mathbf{x}}_p^i = \{\hat{f}_p^i(u_q)\}_{u_q\in\mathcal{U}}$ which is the density function evaluated at some specific points $u_q \in \mathcal{U}$; $\mathcal{U} \subset \mathbb{R}$. According to this the feature vector $\mathbf{f}_{x^i}$ is expressed by:

$$\mathbf{f}_{x^i} = [\breve{\mathbf{x}}_1^i \bullet \breve{\mathbf{x}}_2^i \bullet \ldots \bullet \breve{\mathbf{x}}_P^i] \tag{3.16}$$

Local coefficients as features: It assumes scaling functions with different parameters k for the probability domain subpatterns $\breve{\mathbf{x}}_p^i$ of each time series in a data set. If in Equation (3.12) and Equation (3.13) instead of using a generic $k \in \mathcal{K}$, a specific $k_p \in \mathcal{K}_p = \{-(2n_\phi - 1), \ldots, 2^{j_0}\} \subset \mathbb{Z}$ is considered then the vector $\mathbf{k}_p^i = \mathcal{K}_p$ corresponding to each $\bar{\mathbf{x}}_p^i$ is additionally included on the feature vector which has the form

$$\mathbf{f}_{x^i} = [\mathbf{c}_1^i \bullet \ldots \mathbf{c}_P^i \bullet \mathbf{k}_1^i \bullet \ldots \bullet \mathbf{k}_P^i] \tag{3.17}$$

3.4.4.2 Features and Normalisation Strategies for SGDG

Regarding SGDG, no normalisation strategy is required, however two different alternatives for the features are suggested.

Parameters as features: This strategy involves constructing feature vectors directly from the sets of means, covariances and mixture weights of each probability domain subpattern $\breve{\mathbf{x}}_p^i$. For this purpose, the vectors $\boldsymbol{\mu}_p^i = \{\hat{\mu}_{p,k}^i\}_{k \in \mathcal{K}}$, $\boldsymbol{\Sigma}_p^i = \{\hat{\Sigma}_{p,k}^i\}_{j \in \mathcal{K}}$ and $\boldsymbol{w}_p^i = \{\hat{w}_{p,k}^i\}_{k \in \mathcal{K}}$ are defined. Then, for each subpattern p the vector $\boldsymbol{\mu}_p^i$ is sorted in an increasing order, in such a way that, $\mu_{p,l}^i < \hat{\mu}_{p,l+1}^i$ with $l \in \{1, 2, \ldots, K-1\}$. Finally $\boldsymbol{\Sigma}_p^i$ and $\boldsymbol{w}_p^i$ are arranged according to resulting order of $\boldsymbol{\mu}_p^i$. By following this strategy, the feature vector for $\mathbf{x}^i$ can be expressed as:

$$\mathbf{f}_{x^i} = [\boldsymbol{\mu}_1^i \bullet \ldots \bullet \boldsymbol{\mu}_P^i \bullet \boldsymbol{\Sigma}_1^i \bullet \ldots \bullet \boldsymbol{\Sigma}_P^i \bullet \boldsymbol{w}_1^i \bullet \ldots \bullet \boldsymbol{w}_P^i] \tag{3.18}$$

Densities as features: Similar to the third strategy for SGDW, it considers the density of $\bar{\mathbf{x}}_p^i$ evaluated at some specific points $u_q \in \mathcal{U}$; $\mathcal{U} \subset \mathbb{R}$. Using $\breve{\mathbf{x}}_p^i = \{\hat{f}_p^i(u_q)\}_{u_q \in \mathcal{U}}$ to denote the vector containing the corresponding values of the density evaluated at some points u_q, the feature vector can be defined by

$$\mathbf{f}_{x^i} = [\breve{\mathbf{x}}_1^i \bullet \breve{\mathbf{x}}_2^i \bullet \ldots \bullet \breve{\mathbf{x}}_P^i] \tag{3.19}$$

Table 3.1 shows a summary of the above mentioned feature and normalisation strategies for both SGDW and SGDG algorithms.

3.4.5 Computational Complexity

In this section the complexity analyses for the proposed SGDW and SGDG time series representations are included. Since these representations comprise two subsequent steps, their complexity can be estimated by considering the complexity of each algorithm involved

TABLE 3.1: Features and normalisation strategies for SGDW and SGDG.

Algorithm	**Strategy**	**Features**	**Normalisation**	**Feature Vector**
SGDW1	Global coefficients as features	Scaling function coefficients	Global	$\mathbf{f}_{x^i} = [\mathbf{c}_1^i \bullet \ldots \bullet \mathbf{c}_P^i]$
SGDW2	Local densities as features	Densities	Local	$\mathbf{f}_{x^i} = [\breve{\mathbf{x}}_1^i \bullet \breve{\mathbf{x}}_2^i \bullet \ldots \bullet \breve{\mathbf{x}}_P^i]$
SGDW3	Local coefficients as features	Scaling function coefficients	Local	$\mathbf{f}_{x^i} = [\mathbf{c}_1^i \bullet \ldots \mathbf{c}_P^i \bullet \mathbf{k}_1^i \bullet \ldots \bullet \mathbf{k}_P^i]$
SGDG1	Parameters as features	Means, co-variances and mixture weights	-	$\mathbf{f}_{x^i} = [\boldsymbol{\mu}_1^i \bullet \ldots \bullet \boldsymbol{\mu}_P^i \bullet \boldsymbol{\Sigma}_1^i \bullet \ldots \bullet \boldsymbol{\Sigma}_P^i \bullet \boldsymbol{w}_1^i \bullet \ldots \bullet \boldsymbol{w}_P^i]$
SGDG2	Densities as features	Densities	-	$\mathbf{f}_{x^i} = [\breve{\mathbf{x}}_1^i \bullet \breve{\mathbf{x}}_2^i \bullet \ldots \bullet \breve{\mathbf{x}}_P^i]$

at each step. Additionally, since different strategies for the features for both SGDW and SGDG representations are proposed, then they have different complexities associated.

Firstly, DWT decomposition has a complexity of $O(N \log N)$ [153]. Regarding the complexity of the selected density estimation algorithm, for linear WDE it is $O(N(2^{j_0} + 2n_\phi)(2n_\phi - 1)^3)$ where the term $N_b = 2^{j_0} + 2n_\phi$ refers to the number of basis functions evaluated at resolution 2^{-j_0} to fully cover the interval $[0, 1]$, n_ϕ denoting the order of the scaling function filter, and r expressing the precision in the evaluation of $\phi(.)$ [96]. For the case of FGM the complexity is $O(4IKN)$ [154], where K is the number of Gaussian functions in the mixture and I is the number of iterations employed.

In this way, for WDE, the complexity of strategies based on coefficients as features (first and third strategies) is the same as the complexity of the density estimation algorithm, that is $O(N(2^{j_0}+2n_\phi)(2n_\phi-1)^3)$. Similarly, the complexity for the first strategy based on FGM, parameters as features, is equal to the complexity of estimating FGM, $\sim O(4IKN)$. On the other hand, for strategies relying on densities as features, which considers the evaluation of the density at Q data points, the complexity is $O((N + Q)(2^{j_0} + 2n_\phi)(2n_\phi - 1)^3)$ for WDE and $O(4IKN + QK)$ for FGM.

Since in the proposed framework a multiresolution strategy is followed, the density estimation step is not applied over the original time series of length N, instead of that, it works with the subpatterns generated by the DWT decomposition which have a reduced length that depends on the decomposition level. Note that as the level increases, the number of data points used by the corresponding density estimation block at that level decreases. By considering that DWT includes an approximation of length $\frac{N}{2^M}$ and a set of details at different resolutions with lengths $\frac{N}{2} + \frac{N}{4} + \cdots + \frac{N}{2^M}$ then the complexity of estimating the density of each element in a DWT structure using WDE is $O(N(2^{j_0} + 2n_\phi)(2n_\phi - 1)^3)$ for the first and the third feature strategies and $O((N+Q(M+1))(2^{j_0}+2n_\phi)(2n_\phi-1)^3)$ for the second strategy. In a similar way, the complexity of all the densities in a DWT structure using FGM is $O(4IKN)$ for the strategy based on coefficients and $O(4IKN+QK(M+1))$.

Finally, by combining the complexity of DWT decomposition and density estimation steps, the overall complexity can be obtained, which for SGDW is given by $O(N(2^{j_0} + 2n_\phi + \log N)(2n_\phi - 1)^3)$ when using coefficients as features and $O((N + Q(M + 1))(2^{j_0} + 2n_\phi + logN)(2n_\phi-1)^3)$ when using densities as features. For SGDG there is a complexity equal to $O((4IK+\log N)N)$ when using parameters as features and $O((4IK+\log N)N+QK(M+1))$ when using densities as features.

3.5 Two Online Algorithmic Instantiations

The online algorithms proposed in this section follow the idea of the SGDs time series representation framework presented in Section 3.3. However, since these algorithms are intended to work with unbounded streaming data, the corresponding primitive's attributes and primitive's relations are, in the online context, time dependent. Hence, a recursive updating strategy is required for the SGDs representations, to obtain a time dependent feature vector, every time a new data item becomes available.

Similar to the SGDs representations for time series, their online incremental counterpart in turn can be divided into two main successive stages, namely: (1) online multiresolution decomposition and (2) online density estimation. In the online multiresolution decomposition stage, incoming streaming data is decomposed into subpatterns at different resolutions using linear filtering methods. In the online density estimation stage, data streams subpatterns are mapped into the probability domain by applying recursive density estimation techniques. Both algorithms rely on the use of RWDE for the density estimation stage and differ in the online multiresolution approach adopted. While the first algorithm, referred to as OSGD-D, is based on an online implementation of DWT, the second algorithm, called OSGD-E, is based on an ensemble of Exponential Weighting Moving Average (EWMA) filters.

Note that the proposed online SGDs algorithms assume a set of N univariate data streams $\mathbf{X} = \{\mathbf{x}^i\}, i \in \mathcal{N} = \{1, 2, \ldots, N\}$, where each $\mathbf{x}^i = \{x^i(t)\}$ is an ordered sequence of real valued observations taken at discrete times $t \in \mathcal{T} = \{1, 2, \ldots\}$. Therefore, the objective of the description task is to extract, every time a new data item becomes available, an updated fixed-length feature vector $\mathbf{f}_{x^i}(t)$ from $x^i(t)$, suitable to perform subsequent data mining tasks, that is, clustering and change detection.

3.5.1 Online Multiresolution Decomposition

The first step in the online SGDs algorithms proposed is the online multiresolution decomposition of streaming data using online filtering concepts. On the one hand, the proposed OSGD-D algorithm, is based on an Online Discrete Wavelet Transform (ODWT) which can be seen as a set of low and high pass Finite Impulse Response (FIR) filters. On the other hand, the proposed OSGD-E algorithm considers a multiresolution implementation of Infinite Impulse Response (IIR) EWMA filters (MREWMA). Note that, since ODWT relies on a bank of FIR filters (wavelet and scaling function filters) it is a sliding window-based approach. In contrast, the MREWMA decomposition method, which is based on IIR concepts, is an exponential window-based technique .

3.5.1.1 Online Discrete Wavelet Transform (ODWT)

The first online multiresolution decomposition proposed is based on DWT whose theoretical foundations where presented in Chapter 2. In [59], Mallat proposed a fast algorithm for computing the wavelet decomposition of signals based on representing the projection of data onto the corresponding basis function as a filtering operation. In this way, convolution with a filter $\tilde{\mathbf{h}}$ represents projection on the scaling function, whereas convolution with a filter $\tilde{\mathbf{g}}$ represents projection on a wavelet. Thus, coefficients at different resolutions are obtained using the following equations

$$\mathbf{a}_m = \tilde{\mathbf{h}} * \mathbf{a}_{m-1} \tag{3.20}$$

$$\mathbf{d}_m = \tilde{\mathbf{g}} * \mathbf{a}_{m-1} \tag{3.21}$$

where $\mathbf{a}_m$ and $\mathbf{d}_m$ are the vectors of scaling and wavelet coefficients associated with resolution 2^{-m}, with the symbol $*$ denoting convolution, and m referring to the level of decomposition. Here, the original signal is considered to be at level $m = 0$ and resolution 2^0 whereas the level $m = M$ is associated with the coarsest resolution for the analysis 2^{-M}, here $m, M \in \mathbb{Z}$. Note that the procedure defined by Equation (3.20) and Equation (3.21) has a recursive nature since, at level m, both the vector of scaling coefficients $\mathbf{a}_m$ and the vector of wavelet coefficients $\mathbf{d}_m$ are obtained using the vector of scaling coefficients at a lower level of decomposition $m - 1$.

The online implementation of DWT proposed in this chapter consists in updating DWT coefficients every time a new data item is available. Since data stream applications require the online processing of arriving data then the updating strategy focuses only on updating the most recent DWT coefficients at each resolution. For this, a sliding window approach is followed in which different window sizes are employed at each level of decomposition. Therefore, the length of window is equal to the length of the corresponding filter at that decomposition level.

The proposed approach do not consider any down sampling operation as in [59], and in that sense, different filters $\tilde{\mathbf{h}}$ and $\tilde{\mathbf{g}}$ are used at each level of decomposition. These filters are directly applied over the vector $\mathbf{x}_w^i$ containing the data items covered by the sliding window for data stream $\mathbf{x}^i$. Since these filters now depend on the level, they are denoted as $\tilde{\mathbf{h}}_m$ and $\tilde{\mathbf{g}}_m$, respectively. According to this, and in the context of the proposed framework, Equation (3.20) and Equation (3.21) can be rewritten using matrix notation as:

$$a_m^i(t) = \tilde{\mathbf{h}}_m \mathbf{x}_{w_m}^i \tag{3.22}$$

$$d_m^i(t) = \tilde{\mathbf{g}}_m \mathbf{x}_{w_m}^i \tag{3.23}$$

where $a_m^i(t)$ and $d_m^i(t)$ are the outputs of the filters $\tilde{\mathbf{h}}_m$ and $\tilde{\mathbf{g}}_m$, respectively. Note that $a_m^i(t)'s$ and $d_m^i(t)'s$ are the so called approximation and detail coefficients of $\mathbf{x}_{w_m}^i$. Here, at level m, the filters are defined by

$$\tilde{\mathbf{h}}_m = [h_m(0), \ldots, h_m(L_m - 1)] \tag{3.24}$$

$$\tilde{\mathbf{g}}_m = [g_m(0), \ldots, g_m(L_m - 1)] \tag{3.25}$$

where L_m is the length of the filter and the window covering the most recent L_m data items at level m is expressed by:

$$\mathbf{x}^i_{w_m} = [x^i(t), \ldots, x^i(t - L_m + 1)]^T \tag{3.26}$$

In Equation (3.22) and Equation (3.23) the most recent L_m data items of a data stream $\mathbf{x}^i$, which are covered by a sliding window $\mathbf{x}^i_{w_m}$, are decomposed into scaling and wavelet coefficients of resolution m.

According to the SGDs framework introduced in Chapter 3, and following inverse DWT concepts, the online time domain subpattern of $\mathbf{x}^i$ inside the window $\mathbf{x}^i_{w_M}$ corresponding to level M associated with the coarsest resolution 2^{-M} can be expressed as:

$$\mathbf{x}^i_M = a^i_M(t)\mathbf{h}_M \tag{3.27}$$

The time domain subpatterns related to higher resolutions can be defined as:

$$\tilde{\mathbf{x}}^i_m = d^i_m(t)\mathbf{g}_m \tag{3.28}$$

where $\mathbf{h}_m$ and $\mathbf{g}_m$ denote reconstruction filters and the subpatterns $\mathbf{x}^i_M$ and $\tilde{\mathbf{x}}^i_m$ have the form $\mathbf{x}^i_M = [x^i_M(t), \ldots, x^i_M(t - L_m + 1)]^T$, $\tilde{\mathbf{x}}^i_m = [\tilde{x}^i_m(t), \ldots, \tilde{x}^i_m(t - L_m + 1)]^T$. Figure 3.3 shows both the analysis as well as the synthesis digital filter structures for ODWT. Note however, that in the proposed framework only the part related to the analysis is used. Hence, the synthesis block is shown only for completeness, and because it helps to understand the structural concepts behind the proposed ODWT algorithm. It is important to highlight from Figure 3.3 the time variant nature of $\mathbf{x}^i_{w_m}$ as well as of its corresponding subpatterns $\mathbf{x}^i_M$ and $\mathbf{x}^i_m$'s, which now depend on t.

The structural organisation of the proposed online multiresolution decomposition, in which the data stream pattern $\mathbf{x}^i$ is constructed by combining its corresponding subpatterns at different resolutions, becomes more evident when the synthesis block of Figure 3.3 and the following formal expression are considered:

$$x^i(t) = x^i_M(t) + \sum_m \tilde{x}^i_m(t) \tag{3.29}$$

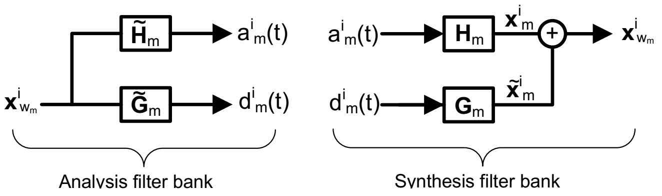


FIGURE 3.3: ODWT filter banks.

In the SGDs framework, coefficients $a^i_M(t)$ and $d^i_m(t)$ are considered condensed representations of $\mathbf{x}_M$ and $\tilde{\mathbf{x}}_m$, since they contain all the information corresponding to resolutions 2^{-M} and 2^{-m}, respectively. In this way, the set of P online multiresolution wavelet domain subpatterns of $\mathbf{x}^i$ at time stamp t can be expressed as:

$$\{\bar{x}^i_p(t)\}_{p\in\mathcal{P}} = \{a^i_M(t), d^i_M(t), d^i_{M-1}(t), \ldots, d^i_1(t)\} \tag{3.30}$$

where $\{\bar{x}^i_p(t)\}_{p\in\mathcal{P}}$ with $\mathcal{P} = \{1, 2, \ldots, P\}$ and $P = M+1$. Note that since one subpattern at resolution 2^M and M subpatterns at higher resolution are involved then the total number of subpatterns P is equal to $M+1$.

Note that from the point of view of digital signal processing, both $\tilde{\mathbf{h}}_m$ and $\tilde{\mathbf{g}}_m$ of Equation (3.24) and Equation (3.25) are in fact Weighted Moving Average (WMA) FIR filters. In that sense, Equation (3.22) and Equation (3.23) can be rewritten as $a_m(t) = \sum_{k=0}^{L_m-1} h_m(k)x(t-k)$ and $d_m(t) = \sum_{k=0}^{L_m-1} g_m(k)x_m(t-k)$, where $a_m(t)$ and $d_m(t)$ are the outputs of the scaling filter $\tilde{\mathbf{h}}_m$ and the wavelet filter $\tilde{\mathbf{g}}_m$, respectively.

The pseudocode for the proposed ODWT approach is shown in Algorithm 3.1. The updating strategy followed is depicted in Figure 3.4. A diagram showing how filters $\tilde{\mathbf{h}}_m$ and $\tilde{\mathbf{g}}_m$ are arranged in the proposed online DWT structure is presented in Figure 3.3

Algorithm 3.1: ODWT ($\mathbf{x}^i_w$, $\tilde{\mathbf{h}}_M$, $\{\tilde{\mathbf{g}}_m\}_{m\in\mathcal{M}}$,$M$)

```
Input: $\mathbf{x}^i_w$: A vector with the $L_m$ most recent data items available from data stream $\mathbf{x}^i$; $\tilde{\mathbf{h}}_M$:
        Scaling filter at level $M$ associated with resolution $2^{-M}$; $\{\tilde{\mathbf{g}}_m\}_{m\in\mathcal{M}}$: Set of wavelet filters;
        $M$: Number of decomposition levels.
Output: $a^i_M(t)$: The most recent scaling function coefficient for $x^i(t)$ at decomposition level $M$;
        { $\mathrm{d}^i_m(t)\}_{m\in\mathcal{M}}$: Set of the most recent wavelet coefficients for $x^i(t)$ at decomposition
        levels $m \in \{1, \ldots, M\}$.
for $m \leftarrow 1$ to $M$ do
    $d^i_m(t) = \tilde{\mathbf{g}}_m\mathbf{x}^i_w$
$a^i_M(t) = \tilde{\mathbf{h}}_M\mathbf{x}^i_w$
```

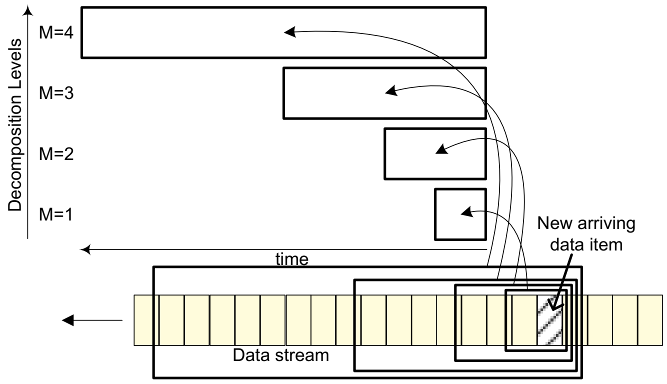


FIGURE 3.4: Updating strategy for the proposed ODWT.

3.5.1.2 Multiresolution Exponentially Weighted Moving Average (MREWMA)

The second online multiresolution decomposition strategy proposed relies on the use of exponential discounting concepts, and its main idea is the construction of IIR filters with similar frequency response than the bank of wavelet FIR filters in a DWT structure. The IIR filter here considered is the so called EWMA filter[2], which is a recursive low-pass IIR filter where the contribution of past observations decreases exponentially as more observations become available.

Note that the EWMA filter represents data at a single resolution. Hence, in order to construct a EWMA-based multiresolution structure, an ensemble of these linear filters, each of them with a different cutoff frequency, is required. For this purpose, a methodology similar to the one used for DWT is followed. Using $m \in \mathcal{M} = \{1, \ldots, M\}$, $\mathcal{M} \subset \mathbb{Z}$ to denote the decomposition level associated with resolution 2^{-m}, considering the original data stream to be at $m = 0$, and relating the level at $m = M$ to the coarsest resolution 2^{-M}, then the output of the filter at level m can be defined by:

$$x^i_m(t) = \alpha_m x^i_{m-1}(t) + (1 - \alpha_m) x^i_m(t-1) \tag{3.31}$$

where $0 < \alpha_m < 1$. In Equation (3.31) each output $x^i_m(t)$ is the weighted average of the previous output of the filter $x^i_m(t-1)$ and the most recent data item at previous decomposition level $x^i_{m-1}(t)$. Note here that $x^i_0(t)$ is the most recent data item of the original data stream.

[2] In financial data analysis, this filter is also known as exponential smoothing [155], while in digital signal processing, it is often called the alpha filter [156].

According to Equation (3.31) the weights assigned to previous values of $x^i_m(t)$ decrease exponentially depending on the value of parameter α_m. Where the choice of α_m depends on the importance given to the current data item as compared to previous data items. Then by choosing different values of α_m for each decomposition level, a multiresolution weighting scheme can be implemented.

Similar to Equation (3.27) the most recent element $x^i_M(t)$ belonging to time domain subpattern of $\mathbf{x}^i$ corresponding to the level M associated with the coarsest resolution 2^{-M} is expressed as:

$$x^i_M(t) = \alpha_M x^i_{M-1}(t) + (1-\alpha_M)x^i_M(t-1) \tag{3.32}$$

Then, by considering the difference between consecutive approximations, subpatterns at successive resolutions, $\tilde{x}^i_m(t)$ can be expressed by:

$$\tilde{x}^i_m(t) = x^i_{m-1}(t) - x^i_m(t) \tag{3.33}$$

where $m \in \mathcal{M} = \{1, \ldots, M\}$, $\mathcal{M} \subset \mathbb{Z}$.

The structural organisation of the proposed MREWMA becomes evident when, at a given time stamp t, the sum of the online multiresolution subpatterns of $\tilde{x}^i_m(t)$ and $x^i_M(t)$ is considered. This sum which is equals to $x^i(t)$ can be formally expressed as:

$$x^i(t) = x^i_M(t) + \sum_m \tilde{x}^i_m(t) \tag{3.34}$$

Therefore, the set of P multiresolution subpatterns of $x^i(t)$ can be expressed by the following equation:

$$\{\bar{x}^i_p(t)\}_{p\in\mathcal{P}} = \{x^i_M(t), \tilde{x}^i_M(t), \tilde{x}^i_{M-1}(t) \ldots, \tilde{x}^i_1(t)\} \tag{3.35}$$

where $\{\bar{x}^i_p(t)\}_{p\in\mathcal{P}}$ with $\mathcal{P} = \{1, 2, \ldots, P\}$ and the total number of subpatterns $P = M + 1$.

Using the z-transform the transfer function of this filter can be expressed as $H_{\alpha_m}(z) = \frac{\alpha_m}{1-(1-\alpha_m)z^{-1}}$. In time domain the input-output relation of this filter can be expressed as $h_{\alpha_m}(t) = \alpha_m(1-\alpha_m)^n u(t)$. Then, Equation (3.31) can be rewritten as $x^i_m(t) = \sum_{k=-\infty}^{\infty} h_{\alpha_m}(k)x^i_{m-1}(t-k)$.

The updating strategy and the corresponding pseudocode for the proposed MREWMA are shown in Figure 3.5 and Algorithm 3.2, respectively. Note that in Figure 3.5, the colour indicates the importance given to new data, where the lighter the colour the higher the importance.

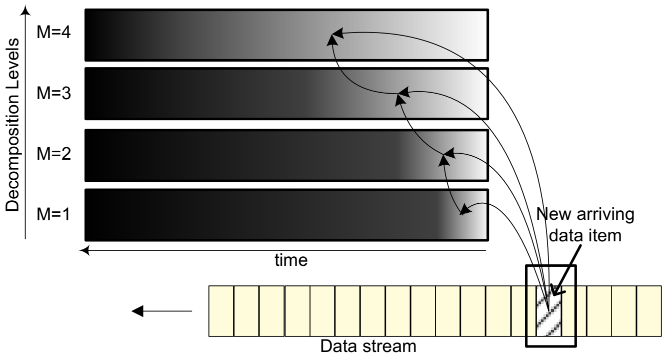


FIGURE 3.5: Updating strategy for the proposed MREWMA.

Algorithm 3.2: MREWMA ($x^i(t)$, $\{\alpha_m\}_{m\in\mathcal{M}}$,$M$)

```
1
Input: $x^i(t)$: the most recent data item available from data stream $\mathbf{x}^i$; $\{\alpha_m\}_{m\in\mathcal{M}}$: Set of
        discounting parameters; $M$: Number of decomposition levels.
Output: $x^i_M(t)$: approximation for $x^i(t)$ at resolution $M$; $\{\tilde{x}^i_m(t)\}_{m\in\mathcal{M}}$: Set of details for $x^i(t)$
        at resolutions $m \in \mathcal{M}$.
$x^i_0(t) = x^i(t)$
for $m \leftarrow 1$ to $M$ do
    $x^i_m(t) = \alpha_m x^i_{m-1}(t) + (1-\alpha_m)x^i_m(t-1)$
    $\tilde{x}^i_m(t) = x^i_{m-1}(t) - x^i_m(t)$
```

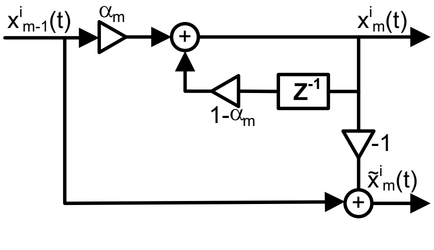


FIGURE 3.6: Online EWMA.

For the estimation of each α_m in the MREWMA algorithm an optimisation procedure is used. This procedure minimises the following cost function:

$$\underset{\alpha_m \in (0,1)}{\arg\min} \left\{ \left| \omega_c^{\tilde{\mathbf{h}}_m} - \omega_c^{EWMA(\alpha_m)} \right| \right\} \tag{3.36}$$

where $\omega_c^{\tilde{\mathbf{h}}_m}$ and $\omega_c^{EWMA(\alpha_m)}$ denote the normalised cutoff frequencies for the filter $\tilde{\mathbf{h}}_m$ and the EWMA filter with parameter α_m, respectively.

The objective of Equation (3.36) is to find EWMA filters with similar frequency response than the ones produced by the use of the scaling function filters $\tilde{\mathbf{h}}_m$ of the ODWT approach. For this purpose the optimisation criterion of Equation (3.36) is used to find the value of α_m that produces a filter with similar cutoff frequency than the scaling function filter at decomposition level m associated with resolution 2^{-m}. Note that in the proposed MREWMA, only scaling function filters ar required, as it can be seen in Figure 3.6, where its corresponding filter bank structure is depicted.

The frequency response for the *db*1 scaling function filter and the corresponding EWMA approximations is shown in Figure 3.7. Additionally, Table 3.2 includes the corresponding α_m's, estimated following the procedure of Equation (3.36), for the first six levels of decomposition m of the first six wavelets from the Daubechies family.

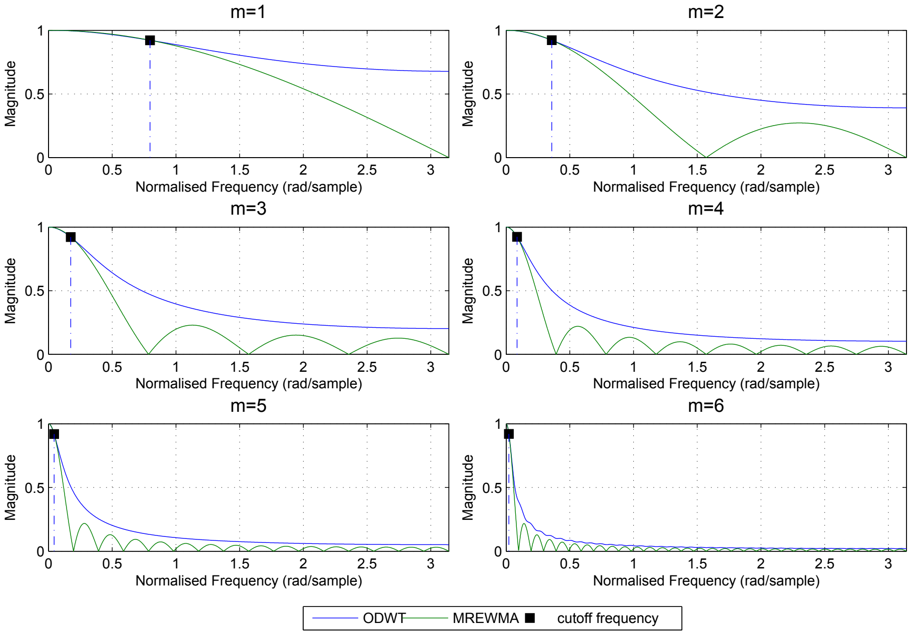


FIGURE 3.7: EWMA filters with similar frequency response than the *db*1 scaling function filters of ODWT for different levels m.

Figure 3.8 shows the corresponding results when the traditional DWT and the proposed ODWT and MREWMA algorithms are applied to an example non stationary data stream.

Table 3.2: EWMA filters with similar cutoff frequency than scaling function filters from the wavelet Daubechies family.

Wavelet	Decomposition level	**Cutoff frequency of scaling function filter** (rad/π)	α_m **of corresponding EWMA Filter**
*db*1	1	0.7964	0.8079
	2	0.3581	0.5617
	3	0.1744	0.3365
	4	0.0864	0.1871
	5	0.0440	0.0973
	6	0.0220	0.0386
*db*2	1	1.0352	0.8650
	2	0.5074	0.6180
	3	0.2529	0.4475
	4	0.1272	0.2613
	5	0.0628	0.1378
	6	0.0314	0.0691
*db*3	1	1.1373	0.8815
	2	0.5655	0.7120
	3	0.2827	0.4818
	4	0.1414	0.2833
	5	0.0707	0.1570
	6	0.0346	0.0786
*db*4	1	1.1985	0.8903
	2	0.5985	0.7294
	3	0.2985	0.5001
	4	0.1492	0.2997
	5	0.0754	0.1646
	6	0.0377	0.0902
*db*5	1	1.2378	0.8948
	2	0.6189	0.7393
	3	0.3094	0.5115
	4	0.1539	0.3047
	5	0.0770	0.1672
	6	0.0393	0.0892
*db*6	1	1.2676	0.8983
	2	0.6346	0.7465
	3	0.3173	0.5203
	4	0.1587	0.3123
	5	0.0785	0.1701
	6	0.0393	0.0892

Note that DWT results depicted in Figure 3.8 were obtaining by using an offline batch-based processing approach which means that the decomposition process considered the whole data stream and was applied once. On the contrary, results for ODWT and MREWMA were obtaining by following an online approach. There are two main observations from this figure. The first one is that, both ODWT and MREWMA algorithms are able to separate information at different scales. The second observation is that results regarding detail coefficients for the level of decomposition $m = 3$ are the ones more dissimilar respect to the result provided by the use of DWT. This is due to the fact that in the offline implementation, at a given level of decomposition, each data sample is involved in the computation of one single coefficient. On the contrary, in the online context, each sample may be related to consecutive coefficients depending on the size of the wavelet or scaling filter (for ODWT) or depending on the value of the corresponding parameter α_m (for MREWMA). As a result, the decomposition provided by the proposed algorithms is redundant, and this redundancy increases as the level of decomposition also increases. Note however that redundancy could be an advantage in some cases, specially when dealing with missing information.

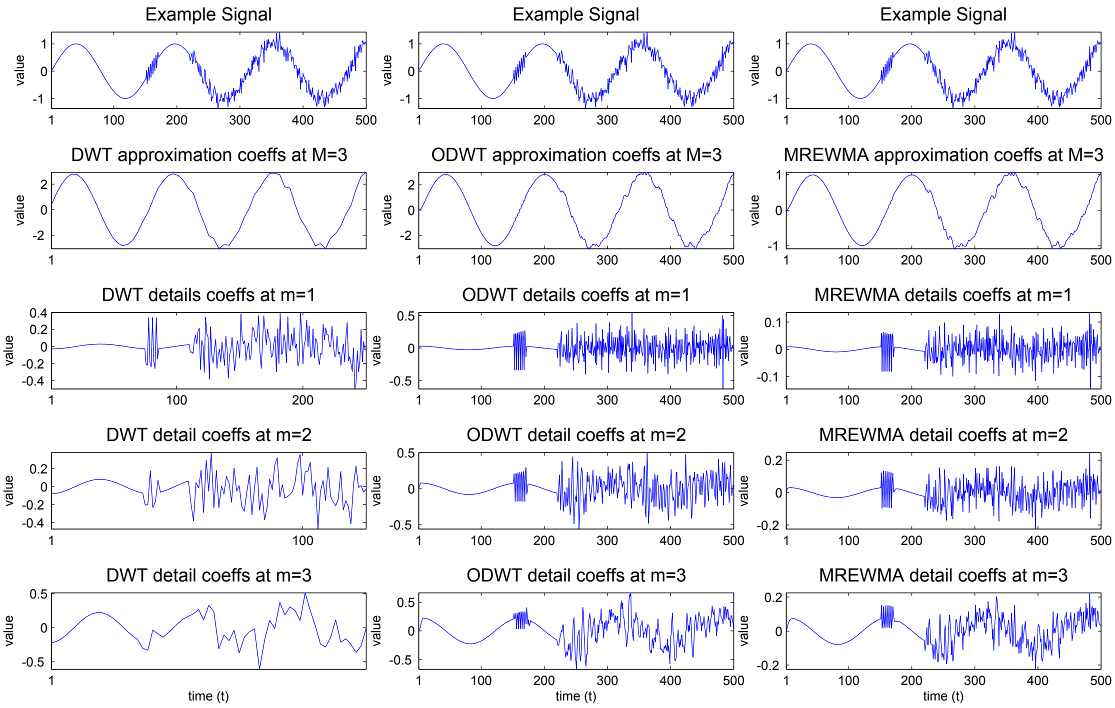


FIGURE 3.8: A comparison between DWT and the proposed ODWT and MREWMA algorithms for a given example signal using *db*1 as wavelet.

3.5.2 Online Density Estimation

The second stage in the proposed online SGDs algorithms consists of mapping of online multiresolution subpatterns obtained using either ODWT or MREWMA into the probability domain by estimating their corresponding probability densities. For this purpose the RWDE suggested in [157] is used. Note that this estimator is based on EWMA concepts and, in that sense, it relies on the use of an exponential discounting strategy for the updating estimator's coefficients.

Since in data streams applications are constrained by computational restrictions, this work considers the simplest RWDE, which is a linear estimator defined by:

$$\hat{f}_p^i(u_q) = \sum_k \hat{c}_{p,j_0,k}^i \phi_{j_0,k}(u_q) \tag{3.37}$$

where $\phi_{j_0,k}(u_q) = 2^{-j_0/2}\phi(2^{-j_0}u_q - k)$ is the scaling function associated with the *base resolution* 2^{-j_0}, $j_0 \in \mathbb{Z}$ and $k \in \mathcal{K} \subset \mathbb{Z}$. Here $u_q \in \mathcal{U} = \{u_1, \ldots, u_Q\}$; $U \subset \mathbb{R}$ denotes a set of data points in which the density $\hat{f}_p^i$ is evaluated.

In Equation (3.37), the estimator's coefficients $\hat{c}_{p,j_0,k}^i(t)$ are recursively updated as new data items arrive according to the equation:

$$\hat{c}_{p,j_0,k}^i(t) = (1-\theta)\hat{c}_{p,j_0,k}^i(t-1) + \theta\phi_{j_0,k}(\bar{x}_p^i(t)) \tag{3.38}$$

where θ is the estimator's discounting parameter that controls the emphasis assigned to new data respect to the older one. Note that in this thesis the notation used for the discounting parameter is different from the originally presented in [157]. Here, the term $1-\theta$ is the one associated with the weighting value for $\hat{c}_{p,j_0,k}^i(t-1)$. By following this strategy θ can be directly related to a sliding window of length $w = 1/\theta$. In this way, if for example, the estimation of the density related to the 100 most recent data items available is required, then θ is set to $1/100$. Moreover, it is important also to highlight that in Equation (3.38) it is normally assumed that $\bar{x}_p^i(t)$ takes values within the interval $[0,1]$, in this way the set of translation indices becomes $\mathcal{K} = \{-(2n_\phi - 1), \ldots, 0, \ldots, 2^{j_0}\}$ with n_ϕ denoting the length of the filter ϕ_{j_0}. For further details about the online implementation of online WDE's the reader is referred to [96, 157].

Recall that for the SGDs framework introduced in Section 3.3, at time stamp t, the online probability domain subpattern $\breve{\mathbf{x}}_p^i(t)$ related to the p-th decomposition level could be either the density $\hat{f}_p^i(u_q)$ evaluated at some points u_q, or could be the set of N_b scaling function coefficients (where $N_b = 2n_\phi + 2^{j_0}$) which are a condensed representation of the former. By taking into account that in the online context the algorithm with the lowest computational complexity is preferred, then, for the online OSGDs algorithms the corresponding RWDE coefficients from time domain subpattern $\bar{x}_p^i(t)$ are selected as the online probability domain subpatterns for the data stream $\mathbf{x}^i$.

3.5.2.1 Normalisation and Feature Vector Integration

The SGDs online alternatives also require the normalisation of input observations. In order to restrict the evaluation of basis functions to the interval $[0, 1]$ in the RWDE stage, a similar approach than the one used in Section 3.4.4 for SGDW algorithms is followed. Specifically, each online probability domain subpattern $\breve{x}^i_p(t)$ is then normalised according to the equation $\hat{x}^i_p(t) = (\breve{x}^i_0(t) - b^{lower}_p)/r$; where $\hat{x}^i_p(t)$ is a normalised data point and the interval r is defined by $r = b^{upper}_p - b^{lower}_p$. Here, b^{lower}_p and b^{upper}_p are the the upper and the lower observation bounds related to the smallest and the greatest observation that can be included in the RWDE density estimate. Note that all those data point outside the interval r would be ignored by the RWDE algorithm since they are outside the support of the corresponding basis functions.

For the online context global normalisation is choosen since, this strategy presents the lowest computational time. This strategy considers the same bounds b^{upper}_p and b^{lower}_p for the online probability domain subpatterns $\breve{x}^i_p(t)$ at the same decomposition level of all the data streams $\mathbf{x}^i$ in a data set. Hence, it is suggested to set $b^{lower}_p = \mu_{\mathbf{X}_p} - 3\sigma_{\mathbf{X}_p}$ and $b^{upper}_p = \mu_{\mathbf{X}_p} + 3\sigma_{\mathbf{X}_p}$, where $\mu_{\mathbf{X}_p}$ and $\sigma_{\mathbf{X}_p}$ denotes the mean and standard deviation of the set $\{x^i_p(t)\}_{i \in \mathcal{N}, t \in \mathcal{T}}$. By following this strategy the same set of basis functions $\{\phi_{j_0,k}\}_{j_0 \in \mathbb{Z}, k \in \mathcal{K}}$ are used in RWDE at each level of decomposition p for all data streams in a data set.

Regarding the features, it is proposed to use the coefficients of the density estimator stage. Formally, by defining a vector of scaling function coefficients $\mathbf{c}^i_p(t) = \{\hat{c}^i_{p,j_0,k}(t)\}_{k \in \mathcal{K}}$, the corresponding feature vector for data stream pattern $\mathbf{x}^i$ at time stamp t is expressed by:

$$\mathbf{f}_{x^i}(t) = [\mathbf{c}^i_1(t) \bullet \ldots \bullet \mathbf{c}^i_P(t)] \tag{3.39}$$

where the symbol $\bullet$ denotes concatenation.

In Figure 3.9 the block diagrams for the proposed OSGD-D and OSGD-E algorithms are depicted. Their corresponding psuedocodes are presented in Algorithm 3.3, for OSGD-D, and Algorithm 3.4, for OSGD-E. It can be noted from Figure 3.9 that both algorithms share the RWDE stage and the way time domain subpatterns are extracted is what makes both techniques different. Since the ODWT stage in OSDG-D is based on FIR concepts the set of the most recent L_m data items available are needed for the extraction of time domain subpatterns. In contrast, for the MREWMA stage in OSGD-E, which relies on IIR concepts, only the most recent data item is required.

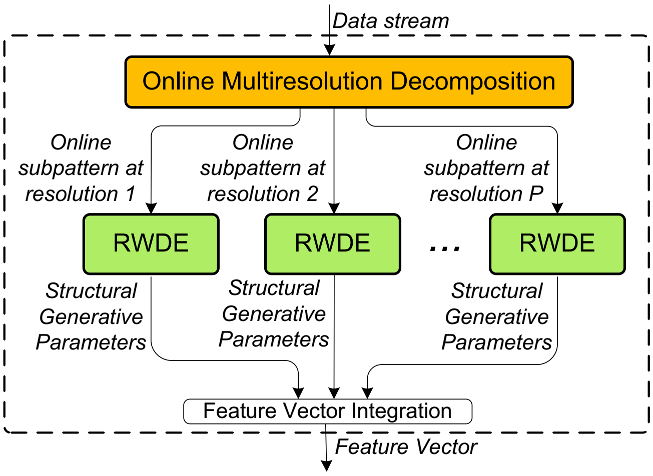

Data stream
Online Multiresolution Decomposition
Online subpattern at resolution 1
Online subpattern at resolution 2
Online subpattern at resolution P
RWDE
RWDE
...
RWDE
Structural Generative Parameters
Structural Generative Parameters
Structural Generative Parameters
Feature Vector Integration
Feature Vector

FIGURE 3.9: Proposed OSGD-D (Online multiresolution decomposition=ODWT) and OSGD-E (Online multiresolution decomposition=MREWMA) algorithms.

Algorithm 3.3: OSGD-D($\mathbf{x}, P$)

1

Input: $\mathbf{x}^i_{w_M} = \{x^i(t-L_M+1), \ldots, x^i(t)\}$: The set of L_M most recent data items from data stream $\mathbf{x}^i$; M: Number of decomposition levels

Output: A feature vector $\mathbf{f}_{x^i}(t)$ for $\mathbf{x}^i$ at time stamp t.

```
1. Decompose $x^i(t)$ using Algorithm 3.1 to obtain, at time stamp $t$, the set of online
   multiresolution subpatterns $\{\bar{x}^i_p(t)\}_{p\in\mathcal{P}}$.
2. Using each $\bar{x}^i_p(t)$ estimate, at time stamp $t$, the corresponding probability domain subpattern
   $\breve{x}^i_p(t)$ using RWDE.
3. For each $\mathbf{x}^i$ obtain, at time stamp $t$, the corresponding feature vector using Equation 3.39.
```

Algorithm 3.4: OSGD-E($\mathbf{x}, P$)

1

Input: $x^i(t)$: The most recent data item available from data stream $\mathbf{x}^i$; M: Number of decomposition levels

Output: A feature vector $\mathbf{f}_{x^i}(t)$ for $\mathbf{x}^i$ at time stamp t.

```
1. Decompose $x^i(t)$ using Algorithm 3.2 to obtain, at time stamp $t$, the set of online
   multiresolution subpatterns $\{\bar{x}^i_p(t)\}_{p\in\mathcal{P}}$.
2. Using each $\bar{x}^i_p(t)$ estimate, at time stamp $t$, the corresponding probability domain subpattern
   $\breve{x}^i_p(t)$ using RWDE.
3. For each $\mathbf{x}^i$ obtain, at time stamp $t$, the corresponding feature vector using Equation 3.39.
```

Figure 3.10 shows the corresponding OSGD-E and OSGD-D representations for a given non stationary data stream, which presents three different temporal behaviours: 1) Sinusoidal behaviour (from time stamp 1 to 1000) 2) Sinusoidal behaviour plus high frequency Gaussian noise (from time stamp 1001 to 2000) and 3) Gaussian noise behaviour (from time stamp 2001 to 3000). The first aspect that is important to highlight from Figure 3.10 is that the two proposed representations are very similar, with a correlation 97.69%

between them. It can also be observed that probability domain subpatterns associated with different resolutions are able to capture different aspects of the studied stream. For instance, for both techniques, the probability domain subpattern 1, is capable of modelling the properties of the generation process at low frequencies while the probability domain subpattern 2 is useful to capture high frequency characteristics.

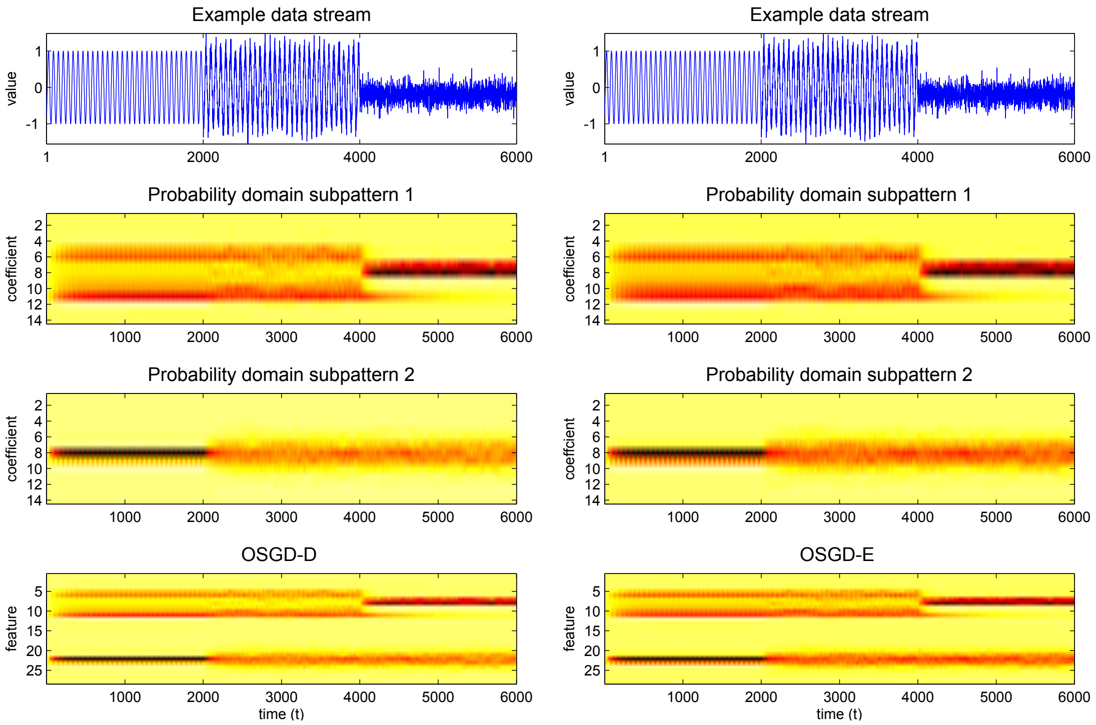


Figure 3.10: OSGD-D and OSGD-E representations for an example data stream.

3.5.3 Computational Complexity

In this section the complexity analysis of the proposed OSGD-D and OSGDG-E algorithms is presented. Since these online data stream representations comprise two subsequent stages, their complexity can be estimated by considering the complexity of each algorithm involved at each stage.

Regarding the multiresolution stage, the proposed ODWT algorithm requires in total $(L_1 + L_2 + \ldots + L_M) + L_M \approx 3L_M$ multiplications resulting in a complexity similar to $O(3L_M)$, where L_M is the length of the filter at the coarsest resolution of the analysis 2^{-M}. On the other hand, MREWMA involves 3 multiplications per coefficient, and since there are $2n_\phi + 2^m$ coefficients at the level of decomposition m, then its corresponding complexity for the whole multiresolution structure is similar to $O(9(2^{M+1}))$.

The complexity of the density estimation stage, which for both algorithms is based on RWDE, involves the evaluation of $\phi_{j_0,k}(.)$ for each of the $N_b = (2n_\phi + 2^{j_0})$ scaling functions employed using a given j_0. This evaluation relies on the so called Daubechies-Lagarias Algorithm (see Appendix A for more details) where two are the variables involved: the order of the filter n_ϕ, and the accuracy of the algorithm r. Hence, the complexity of evaluating $\phi_{j_0,k}(.)$ for a single scaling function is $O(r(2n_{\phi_0} - 1)^3)$ while the complexity of updating all the estimator's coefficients is $O(rN_b(2n_\phi - 1)^3)$.

By combining the complexity of the above defined multiresolution decomposition and density estimation stages, the complexity of the OSGD-D algorithm is $O(3rL_M N_b(2n_\phi - 1)^3)$ while the complexity of the OSGD-E algorithm is $O(9rN_b(2^{M+1})(2n_\phi - 1)^3)$.

Note that the computational complexity of the proposed OSGD-based algorithms is the same for every arriving data item and it does not depend on the amount of arriving data to be processed.

Regarding the amount of memory, the OSDG-E algorithm only requires to store the last available data item for each level of decomposition m and the corresponding α_m, resulting in a space complexity equal to $2M$. For the case of OSGD-D, the scaling and wavelet filters associated with each decomposition level in the multiresolution decomposition stage as well as the last L_M data items from the stream need to be stored in memory. This results in a space complexity similar to $3L_M$.

3.6 Final Remarks

In this chapter, a novel time series representation framework suitable for classification applications involving time series and data streams was proposed. This representation relies on including generative and structural aspects into a fixed length statistical feature vector to allow the subsequent use of any of the well established decision-theoretic methods in the classification stage. The proposed SGDs representation framework provides a compact structural representation for time series and data stream patterns that captures the generation process of the data at different resolutions. The framework comprises: 1) a multiresolution decomposition stage in which input patterns are decomposed into simpler subpatterns at different resolutions; and 2) a subsequent density estimation stage in which decomposed patterns are mapped to the probability domain.

In addition to this, two off-line and two online algorithms relying on the proposed SGDs representation framework, as well as different strategies for the selection of their corresponding features were also proposed. The off-line algorithms are based on DWT for the multiresolution decomposition stage, and while the first algorithm uses WDE as density estimator the second method is based on FGM to estimate the density.

Regarding online algorithms, the underlying idea was to reformulate the two stages of off-line SGDs in a recursive manner in order to update the SGDs representations as new data items become available. Hence, the online framework involves the online learning of a set of different resolution probability-domain subpatterns for input data. The multiresolution decomposition stage, for the first online algorithm proposed considers an online implementation of DWT, while for the second online algorithm is based on EWMA filters. Both algorithms rely on RWDE for the online density estimation stage.

In this electric age we see ourselves being translated more and more into the form of information, moving toward the technological extension of consciousness.

Marshall Mcluhan,
Understanding Media: The Extensions of Man, MIT Press, (1994)

Chapter 4

Empirical Evaluation of SGDs Algorithms

In this chapter the empirical evaluation for the offline and online algorithmic instantiations of the proposed SGDs framework is presented. The offline algorithms are evaluated in the context of time series classification using benchmark synthetic and real world data. The performance of the online algorithms is assessed in the context of change detection and clustering of data streams.

This chapter is organised as follows. In Section 4.1, the empirical evaluation for the proposed offline SGDs representations is conducted. Section 4.2 presents the corresponding evaluation experiments for the proposed online SGDs algorithms. Final remarks are discussed in Section 4.3.

4.1 Empirical Evaluation Offline of SGDs Algorithms

The empirical evaluation of the proposed SGDs algorithms is divided into two parts. In the first part the computation time of the SGDs algorithmic instantiations is evaluated. In the second part the performance evaluation of the algorithms is carried out on four classification experiments. The first experiment evaluates the SGDs algorithms using a synthetic data set. The second experiment is conducted on rolling element bearing vibration data obtained from the Case Western Reserve University Bearing Data Center (CWRU) [158]. In the third experiment the algorithms are evaluated in a biometrics application using ECG data. Finally, the fourth experiment uses 42 benchmark time series data sets from the University of California Riverside (UCR) time series repository [159].

The procedures used in the assessment consider the fact that algorithms will have different values for their corresponding tuning parameters. For this purpose grids with nodes of the form (W, M, j_0) for SGDW and with nodes of the form (W, M, K) for SGDG are constructed, where each node is a particular combination of the tuning parameters. On the above setting the evaluation is done for each identified node. Specifically, $W = \{db1, db2, db3, bior1.3, bior5.5, coif1, coif3, sym2, sym4\}$ is the wavelet of the DWT decomposition stage, $M \in \{1, \ldots, 6\}$ is the number of decomposition levels, $j_0 \in \{1, \ldots, 6\}$ is the index related to the base resolution for the WDE stage in SGDW algorithms, and $K \in \{1, \ldots, 6\}$ is the number of Gaussian functions in the FGM stage of SGDG approaches. The *Symlet* of order 4 (*Sym*4) is selected as the basis function for the WDE stage in SGDW-based algorithms since it is the least asymmetrical wavelet function [74].

4.1.1 Computation Time Assessment

This assessment focuses on the estimation of the time required by a reference computer[1] to obtain the different variants of the SGDs representations using in turn, different values for their corresponding parameters. For this evaluation a data set comprising 10 stochastic time series of length 1000 generated from the Gaussian distribution $\mathcal{N}(0, 1)$ is used. The computation time required to obtain a given SGDs representation, using the proposed SGDW and SGDG algorithms, with different normalisation and feature strategies is depicted in Figure 4.1. In order to facilitate visualisation, for each algorithm all those nodes from the same wavelet and the same K or j_0 (depending on the case) but with different decomposition levels M are averaged. In this way, a surface for each algorithm with axes consisting of the wavelet and K or j_0 is obtained.

[1] The computer system used to generate the results reported in this section was an Intel i5 2500k with 8 GB of RAM, running on Linux Ubuntu 11.04 and the simulation environment was MATLAB R2010b.

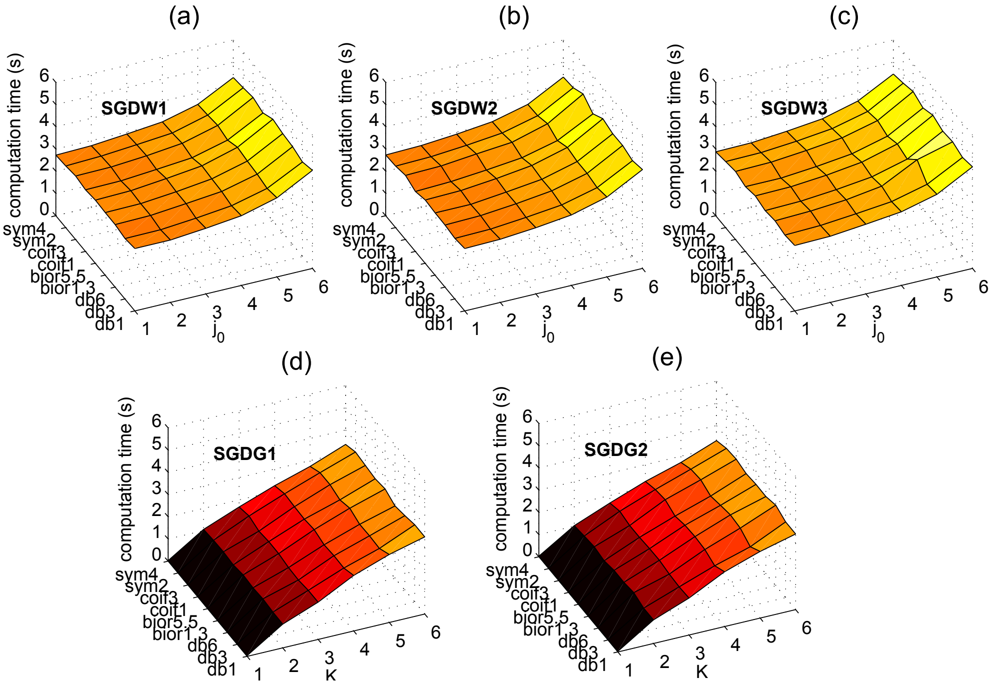


Figure 4.1: Computation time for SGDs representations using an example data set of 10 time series of length 1000, generated from the Gaussian distribution $\mathcal{N}(0,1)$.

The first aspect that is important to highlight from Figure 4.1 is that, since the density estimation technique of SGDW algorithms have higher computational complexity than the one used by the SGDG approaches, the three alternatives for the former are more time consuming than the two versions evaluated for the latter. Note for example that for the nine wavelets evaluated the lowest computation time for SGDW-based algorithms is around 3 seconds, and it is obtained when $j_0 = 1$. Conversely, the computation time when using SGDG-based methods is close to 0.1 seconds for $K = 1$ and it increases as the value of K increases. Furthermore, the slope of the surfaces of Figure 4.1 indicates that in SGDG algorithms there is a high difference of complexity for different values of K. In contrast, the complexity of SGDW-based algorithms suffer a subtle variation when j_0 is modified, specially when $1 < j_0 < 3$. Note that this is an expected result, since the number of basis functions considered in the density estimation stage of SGDW algorithms increases exponentially as j_0 increases.

In Figure 4.2 the computation time obtained using SGDW and SGDG algorithms with different values of M and K or j_0, depending on the case, is averaged for each wavelet. The first observation from Figure 4.2 is that, as it was shown in Figure 4.1, SGDG-based algorithms are around 40% faster than whichever of the three SGDW-based methods. Note also that, in general, the resulting computation time for the proposed algorithms is quite similar using different wavelets.

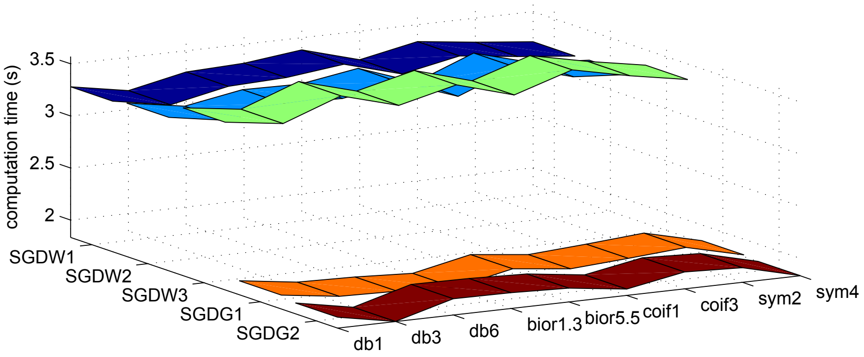


FIGURE 4.2: Averaged computation time for SGDW and SGDG algorithms.

4.1.2 Performance Evaluation of the two SGDs Algorithmic Instantiations

For the four experiments here conducted, the evaluation consists in obtaining the corresponding representation using the proposed SGDs algorithms or a particular benchmark method, for each and every one of time series in a given data set. Each representation is evaluated considering different values of its tuning parameters.

Since the main purpose of these evaluations is the comparison of the description capabilities of the proposed SGDs representations in the context of time series classification, the resulting representations are used as inputs for one simple classifier. The chosen classifier is the 1-NN algorithm, which among the time series classification and clustering community is strongly recommended for comparisons [159]. The evaluation presented in this chapter considers five distance measures. They are the Euclidean Distance (ED), the Standardized Euclidean distance (SE), the Cosine distance (CO), the Chebyshev distance (CH) and the Correlation distance (CR). More details about these distance measures are provided in Table 4.1.

Regarding the benchmark time series representations the evaluation considers the 21 techniques listed in Table 4.2, from which among the most relevant ones methods relying on wavelet and Fourier transforms, Piecewise Aggregate Approximation (PAA) [160], Chebyshev polynomials and Autoregressive models could be cited. Note that PAA is the core of the so called symbolic aggregate approximation algorithm (SAX) [87, 161], which is

TABLE 4.1: Distance measures used in the experiments.

Name	**Abbr**	**Description**	**Formula**
Euclidean distance	ED	The traditional distance between two points given by the Pythagorean formula.	$\text{ED}(y, w) = \sqrt{(y-w)(y-w)^T}$*
Standardised Euclidean distance	SE	Each coordinate difference is scaled by dividing by the corresponding element of the standard deviation.	$\text{SE}(y, w) = \sqrt{(y-w)V^{-1}(y-w)^T}$**
Cosine distance	CO	One minus the cosine of the included angle between points (treated as vectors).	$\text{CO}(y, w) = 1 - \frac{yq^T}{\sqrt{(yy^T)(ww^T)}}$.
Chebyshev distance	CH	It considers the greatest difference along any coordinate dimension.	$\text{CH}(y, w) = \max_i(\lvert y_i - w_i \rvert)$.
Correlation distance	CR	One minus the sample correlation between points (treated as sequences of values).	$\text{CR}(y, w) = 1 - \frac{(y-\bar{y})(w-\bar{w})^T}{\sqrt{(y-\bar{y})(y-\bar{y})^T}\sqrt{(w-\bar{w})(w-\bar{w})^T}}$

*where T denotes the transpose.
**where V is the diagonal matrix of variances at each coordinate.

a popular symbolic time series representation. In Table 4.2, the first four columns refer to the representation number n, the representation technique name, its abbreviation, and the parameter or parameters selected as features for each representation. The last column includes information about the parameters setting of each representation, indicating the tuning parameter and the corresponding range of values considered in the evaluation.

Note that in Table 4.2 three approaches based on DFT and two based on DCT concepts are included. Specifically, DFT refers to the method in which the periodogram is used as feature vector. Regarding DFTW, it refers to the Welch's method of power spectrum estimation in which the Fast Fourier Transform (FFT) is used to estimate the power spectra based on sectioning each time series, obtaining the periodogram for each section, and then averaging these localised periodograms [163]. With respect to DFT2 it relies on transforming a given time series into the frequency domain using DFT and uses the corresponding coefficients as features [164]. For the case of DCT-based algorithms, DCT refers to a DCT-based quantisation method in which thresholding is applied over the DCT coefficients of a given time series, and the time series reconstructed from these thresholded coefficients is used as feature vector. DCT2 is the technique that considers DCT coefficients directly as features.

TABLE 4.2: Benchmark representation techniques used in the experiments.

n	Representation Name	Abbr.	Features	Tuning parameters and Range
1	Raw Data	RAW	Data points	-
2	Statistical Moments	SM	Mean and variance	-
3	Discrete Wavelet Transform	DWT	Wavelet coefficients at different decomposition levels	Decomposition Level: $\{0 - l_{max}\}$
4	Chebyshev Polynomials	CHEB	Polynomial coefficients	Order: $\{1, \ldots, 20\}$
5	Piecewise Aggregate Approximation [160]	PAA	Segments	Segments: $\{1, \ldots, 20\}$
6	ARMA Models	ARMA	Model coefficients	AR Order: $\{1, \ldots, 6\}$, MA Order: $\{1, \ldots, 6\}$
7	ARIMA Models	ARIMA	Model coefficients	AR Order: $\{1, \ldots, 6\}$, MA Order: $\{1, \ldots, 6\}$
8	Multiscale Entropy [162]	MSE	Entropy at different scales	Scales:$\{1, \ldots, 15\}$
9	Fuzzy Multiscale Entropy [1]	FMSE	Entropy at different scales	Scales:$\{1, \ldots, 15\}$
9	Discrete Fourier Transform	DFT	Periodogram data points	Nterms: $\{1, \ldots, 30\}$
11	Discrete Fourier Transform (Welch's Method)	DFTW	Periodogram data points	Nterms: $\{1, \ldots, 20\}$, Segments: $\{1, \ldots, 10\}$
12	Autocorrelation Function	ACF	Correlation at different lags	Lags: $\{1, \ldots, 15\}$
13	Polynomial	POLY	Polynomial coefficients	Order: $\{1, \ldots, 20\}$
14	Single Value Decomposition	SVD	Singular vectors	Singular Vectors: $\{1, \ldots, 20\}$
15	Principal Component Analysis	PCA	Principal Components	Principal Components: $\{1, \ldots, 20\}$
16	Energy of Wavelet Packets	WPE	Energy at different branches of the decomposed tree	Decomposition Level: $\{0, \ldots, l_{max}\}$
17	Statistical Moments of Wavelet Packets	WPS	Mean and variance of coefficients at different branches of the decomposed tree	Decomposition Level:$\{0, \ldots, l_{max}\}$
18	Statistical Moments of Discrete Wavelet Transform	DWTS	Mean and variance of coefficients at different decomposition levels	Decomposition Level: $\{0, \ldots, l_{max}\}$
19	Discrete Cosine Transform	DCT	Transform coefficients	Threshold: $\{0.1 : 0.05 : 0.5\}$
20	Discrete Cosine Transform Version 2	DCT2	Transform coefficients	Nterms: $\{10, 20, 30, \ldots, 100\}$
21	Discrete Wavelet Transform Version 2	DFT2	Transform coefficients	Nterms: $\{10, 20, 30, \ldots, 100\}$

Note also that in Table 4.2 $lmax$ refers to the the maximum allowed level of decomposition in a wavelet-based algorithm. This value is related to the last level for which at least one coefficient is correct [165]. On the other hand, regarding cases in which the length of the time series is less than 100 data points in DCT2 and DFT2 representations the range of the tuning parameters is chosen to be $\{10, 20, 30, \ldots, nmax * 10\}$ with $nmax = l \backslash 10$ where l is the length of the time series and the symbol $\backslash$ denotes the *integer division* operator.

A similar experimental setting is followed for the four experiments. That is, a given percentage of time series from each class is randomly selected as training set and then the 1-NN algorithm is applied to classify the remaining time series. The experiment is repeated

100 times using the averaged classification error over the 100 trials as performance metric. The five distance measures of Table 4.1 are used for the evaluation.

4.1.2.1 Experiment with Synthetic Data Set

The first experiment considers the evaluation of time series representations using a set of 3200 time series of 1000 data items each of them. There are 32 classes in this data set, 100 time series for each class. The experiment consists in predicting the class of the 90% of time series (2880 time series) using for training only the 10% of the time series for each class (320 time series in total).

The synthetic data is generated using the 32 prototype time series of Table 4.3 and Figure 4.3. The generation process of the time series related to a particular prototype involves distorting horizontally the prototype, by randomly modified its given amplitude and by adding noise, as well as distorting vertically the prototype, by introducing a random shift.

TABLE 4.3: Synthetic Time Series Data set.

Name	**Formula**	**Name**	**Formula**
S1	$\sin(\frac{1}{100}t)$	S17	$\frac{5}{10}\sin(\frac{9}{100}t) - \frac{2}{10}\sin(\frac{1}{10}t)$
S2	$\sin(\frac{4}{100}t)$	S18	$\frac{4}{1000}\sum_{i=1}^{t}(\frac{2}{100}\sin(i)) + \frac{2}{1000}\sum_{i=1}^{t}(\frac{8}{100}\sin(i))$
S3	$\frac{1}{2}\sin(\frac{1}{10}t) + \frac{1}{2}$	S19	$\frac{5}{10}\sin(\frac{4}{100}t) - \frac{1}{10}\sin(\frac{1}{10}t)$
S4	$(\frac{5}{100}\sin(t))$	S20	$\frac{3}{10}\sin(\frac{5}{100}t + 50) - \frac{5}{10}\sin(\frac{1}{10}t)$
S5	$-\frac{7}{10}(\frac{15}{1000}\sin(t))$	S21	$\frac{6}{10}(\frac{5}{100}\sin(t))) + \frac{8}{1000}\sum_{i=1}^{t}(\frac{15}{100}\sin(i))$
S6	$\frac{3}{1000}\sum_{i=1}^{t}(\frac{1}{100}\sin(i))$	S22	$\frac{6}{10}(\frac{5}{100}\sin(t) + 23)) - \frac{5}{100}\sum_{i=1}^{t}(\frac{15}{100}\sin(i))$
S7	$\frac{8}{1000}\sum_{i=1}^{t}(\frac{3}{100}\sin(i))$	S23	$\frac{5}{10}(\frac{3}{100}\sin(t)) - \frac{5}{10}\sin(\frac{2}{100}t)$
S8	$\frac{3}{10}\sin(\frac{4}{100}t) + \frac{3}{10}\sin(\frac{5}{100}t) + \frac{3}{10}\sin(\frac{6}{100}t)$	S24	$\frac{6}{10}(\frac{4}{100}\sin(t)) - \frac{3}{10}\sin(\frac{1}{100}t)$
S9	$\frac{8}{10}\sin(\frac{2}{100}t) + \frac{2}{10}\sin(\frac{1}{100}t)$	S25	$\frac{3}{10}\sin(\frac{3}{10}t) + \frac{3}{10}\sin(\frac{2}{10}t) + \frac{3}{10}\sin(\frac{1}{10}t)$
S10	$\frac{5}{10}\sin(\frac{1}{100}t) + \frac{8}{100}\sin(\frac{2}{100}t) + \frac{8}{100}\sin(\frac{4}{100}t)$	S26	$\frac{3}{10}\sin(\frac{3}{10}t) + \frac{3}{10}\sin(\frac{2}{10}t) + \frac{3}{10}\sin(\frac{1}{10}t + 400)$
S11	$\frac{5}{10}\sin(\frac{1}{100}t + 100) + \frac{2}{10}\sin(\frac{5}{100}t + 100) + \frac{8}{100}\sin(\frac{4}{100}t + 100)$	S27	$\frac{3}{10}\sin(\frac{1}{10}t) + \frac{3}{10}\sin(\frac{2}{10}t) + \frac{3}{10}\sin(\frac{1}{10}t + 400)$
S12	$\frac{5}{10}\sin(\frac{1}{100}t + 300) + \frac{3}{10}\sin(\frac{5}{100}t)$	S28	$\frac{3}{10}\sin(\frac{5}{10}t) + \frac{3}{10}\sin(\frac{2}{10}t) + \frac{3}{10}\sin(\frac{1}{10}t + 300)$
S13	$\frac{6}{10}(\frac{1}{100}\sin(t)) + \frac{3}{10}(\frac{1}{10}\sin(t))$	S29	$\frac{3}{10}\sin(\frac{6}{10}t) + \frac{4}{10}\sin(\frac{3}{10}t) + \frac{3}{10}\sin(\frac{2}{10}t + 100)$
S14	$\frac{7}{10}(\frac{1}{100}\sin(t) + 400) + \frac{2}{10}(\frac{5}{100}\sin(t))$	S30	$\frac{7}{10}\sin(\frac{1}{10}t) + \frac{3}{10}\sin(\frac{15}{100}t) + \frac{3}{10}\sin(\frac{4}{10}t)$
S15	$\frac{4}{10}\sin(\frac{5}{100}t) + \frac{4}{10}\sin(\frac{1}{10}t)$	S31	$\frac{5}{10}(\frac{1}{100}\sin(t) + 400) + \frac{4}{10}(\frac{2}{100}\sin(t))$
S16	$\frac{5}{10}\sin(\frac{1}{10}t) + \frac{3}{10}(\frac{2}{10}\sin(t) + 80)$	S32	$\frac{3}{10}(\frac{2}{100}\sin(t) + 400) + \frac{3}{10}(\frac{8}{100}\sin(t)) + \frac{3}{10}(\frac{4}{100}\sin(t))$

*$sgn(X)$ denotes the signum function

Denoting a given prototype time series as $p(t)$ then its corresponding 100 time series are generated according to the following equation:

$$x(t) = (1 + h(t))p(t) + g(t); \tag{4.1}$$

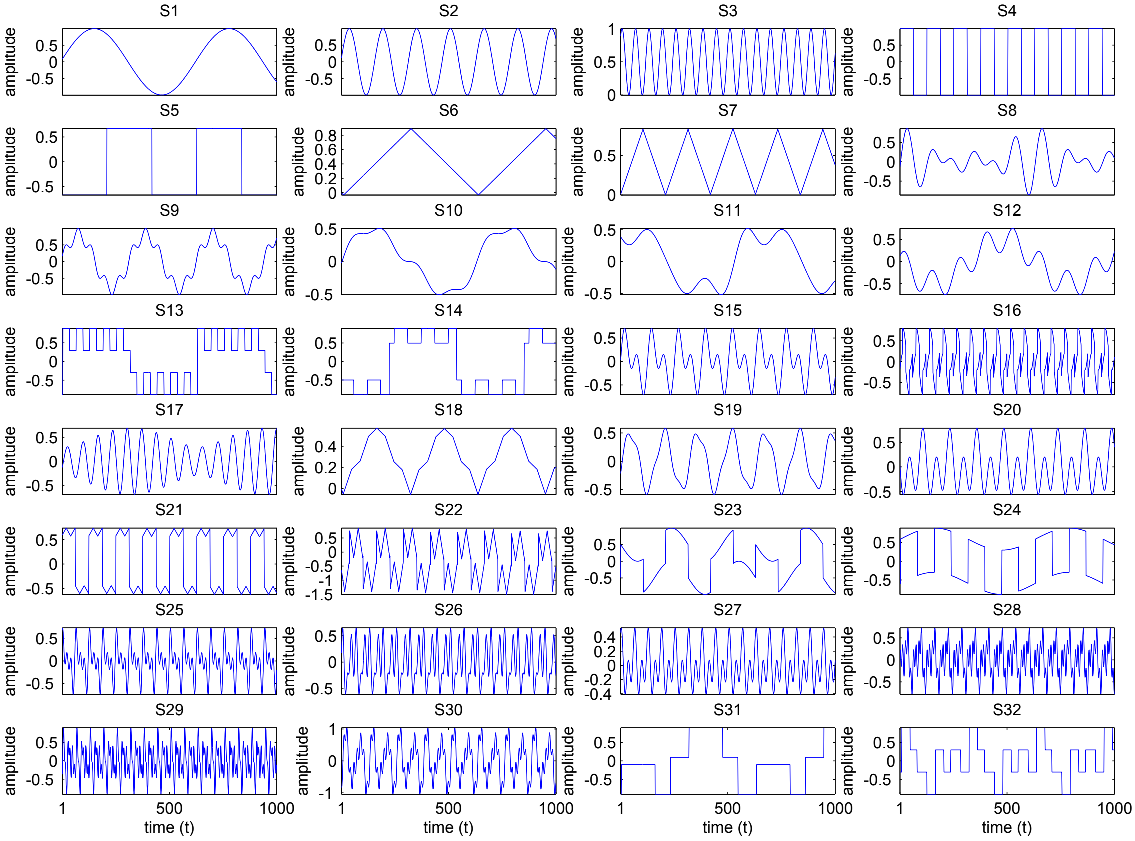


FIGURE 4.3: Prototype time series from the synthetic data set.

with $t = \{1 + q(t), 2 + q(t)..., 1000 + q(t)\}$, where $h(t)$ and $q(t)$ are normally distributed random signals whose points are drawn from $\mathcal{N}(0, 0.01)$, and $\mathcal{N}(0, 25)$, respectively. Regarding $g(t)$, it is a uniformly distributed random noise to produce a Signal-to-Noise Ratio (SNR) of 30 dB.

Results and Discussions on Synthetic Data Assessment: Results for the experiment are summarised in Figure 4.4 where, for each distance measure, the corresponding time series representation are sorted according to the averaged classification error. Note that in Figure 4.4 only the node related to the best wavelet is considered for SGDW and SGDG algorithms. Detailed results are presented in Table A.1 and Table A.2 from the Appendix A.

The most important observation from Figure 4.4 is that the three SGDW-based algorithms proposed report the three best performances. Regarding SGDG-based algorithms, they perform consistently better than 19 out of 21 time series representations investigated, only $DFT2$ and $DFTW$ outperform them. It is important to note that since the time series included in this data set are periodic signals it is expected that a method designed for the analysis of this type of signals such as DFT, DFT2, DFTW, DCT and DCT2 bring the best result. However, as results reported in this section clearly show, the proposed SGDW-based algorithms outperform such representations in this context. Moreover, note that for all the distance measures evaluated, with the exception of SE, the SGDW1 algorithm produces the best representation irrespectively of the chosen wavelet. Furthermore, the performance of SGDW3, with whichever wavelet is better than the 21 benchmark representation studied. Regarding SGDGs algorithms, SGDG1 provided the best result only for the Cosine distance measure. An additional observation is that, both SGDW-based and SGDG-based algorithms offer high performance, in terms of lower classification error, regardless of the distance measure selected for the 1-NN classifier.

In respect to the benchmark representations, it can be observed from Figure 4.4 that in 4 out of the 5 of the distance measures evaluated DFT2 provides the best result. DFTW, which is the best benchmark representation for the Cosine distance, gives the second best result for the remaining four distance measures.

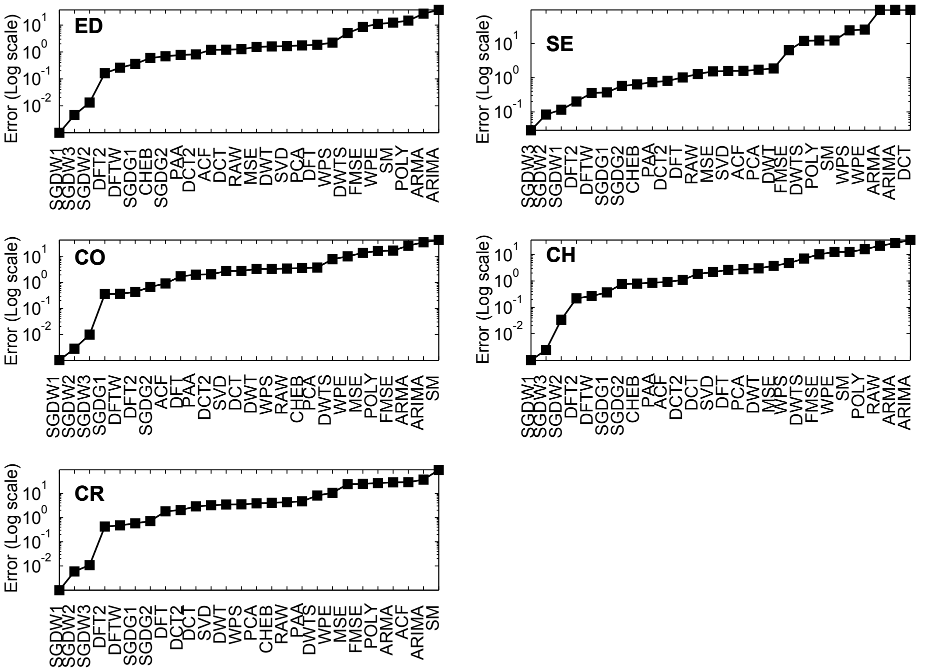


FIGURE 4.4: Errors for different distance measures for synthetic data experiment.

The pixel plots from Figure 4.5 show how the proposed algorithms perform when different levels of decomposition M and different base resolutions j_0 are chosen. In Figure 4.5 a different pixel plot is presented for each wavelet and, in this sense, a set of nine plots are the related to the SGDW1 algorithm (Figure 4.5a) and nine are related to the SGDG1 algorithm (Figure 4.5b). Note also that only results for ED are presented. In these pixel plots the colour is related to the averaged classification error over 100 trials. The darker the colour the smaller the classification error. Note that to facilitate visualisation a different scale is used for each method. From Figure 4.5, the first observation is that low classification errors ar obtained with different wavelets. This result, which applies for both algorithms, implies that regardless of the wavelet employed in the decomposition stage, the proposed SGDs-based algorithms will consistently perform at acceptable levels. The second aspect that is important pointing out is that, for this particular experiment, SGDW1 provides a good performance with a wider range of values for its parameters M and j_0. In contrast, using the SGDG1 algorithm, low classification errors are only obtained when $M = 0$ which refers to the situation in which the density estimation stage directly works with input time series patterns, without any multiresolution decomposition involved.

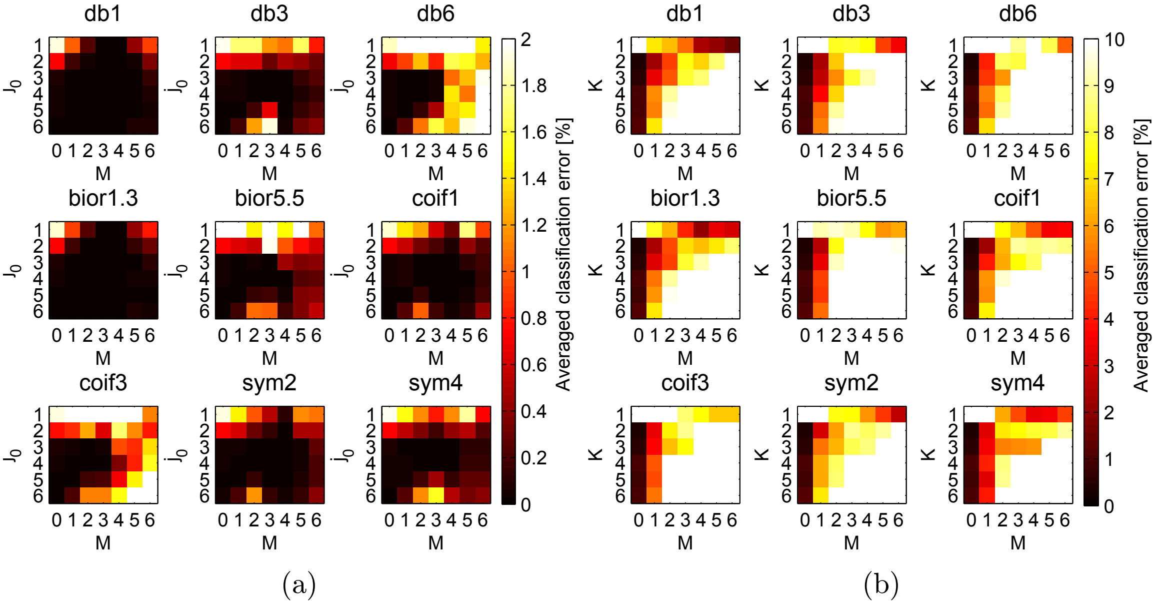


FIGURE 4.5: Pixel plots for SGDW1 (a) and SGDG1 (b) algorithms in Synthetic data experiment using ED as distance measure.

4.1.2.2 Experiment with data from Case Western Reserve University Bearing Data Center

The second experiment considers the evaluation of the proposed time series representations in the context of bearing health condition identification where the diagnosis is based on the analysis of vibration signals in the form of time series. The data set includes 765 time series of length 2048 from 12 bearing conditions. The number of time series per condition is variable. The experiment consists in predicting the class (condition) of the 50% of the time series in each class using the other 50% time series for training.

This experiment is conducted on rolling element bearing vibration data obtained from the Case Western Reserve University Bearing Data Center (CWRU) [158]. The reason for the selection of this data set is three-fold: (1) it is a real world application, (2) it is well documented and publicly available, and (3) traditional solutions are domain dependent. Regarding the last reason, note that proposed SGDs framework is intended to be domain independent.

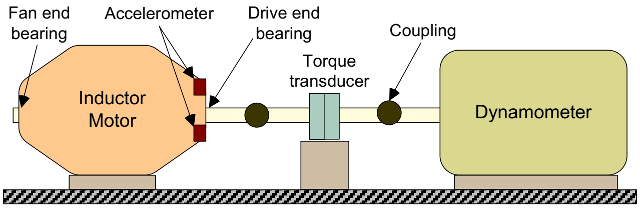


FIGURE 4.6: Scheme diagram of test stand for the experiment [1].

The original CWRU data set considers four different defect sizes, three different bearing locations and the detection scenario in which no load is applied to the induction motor. Then twelve bearing vibration signals for twelve bearing conditions are extracted from the test stand shown in Figure 4.6.

The data set for the experiment is constructed according to the experimental setting proposed in [1] and [166], where data samples of $2,048$ points are extracted from the original signals to form the data set described in Table 4.4. Note that the length of the data samples is selected to be 2,048 points based on the fact that the time spanned by each of them covers about five motor revolutions. Further details about the CWRU vibration data can be found in [158]. Figure 4.7 shows sample time series representative of each of the twelve bearing conditions from Table 4.4.

Results and Discussions on the CWRU Experiment: Figure 4.8 shows, for each distance measure, the averaged classification error reported for each representation ordered in an ascending manner. Note that, for the case of the proposed SGD-based algorithms only the node related to the wavelet that reported the best performance is included. Detailed results for this experiment are included in Table A.3 and Table A.4 from Appendix A.

TABLE 4.4: Description of the CWRU data set [1, 166].

Bearing condition	**Defect size (inches)**	**Number of data samples**	**Condition No.**
Normal	0	119	1
Outer Race (OR)	0.007	59	2
	0.014	59	3
	0.021	59	4
Rolling Element (RE)	0.007	59	5
	0.014	59	6
	0.021	59	7
	0.028	58	8
Inner Race (IR)	0.007	58	9
	0.014	59	10
	0.021	59	11
	0.028	58	12

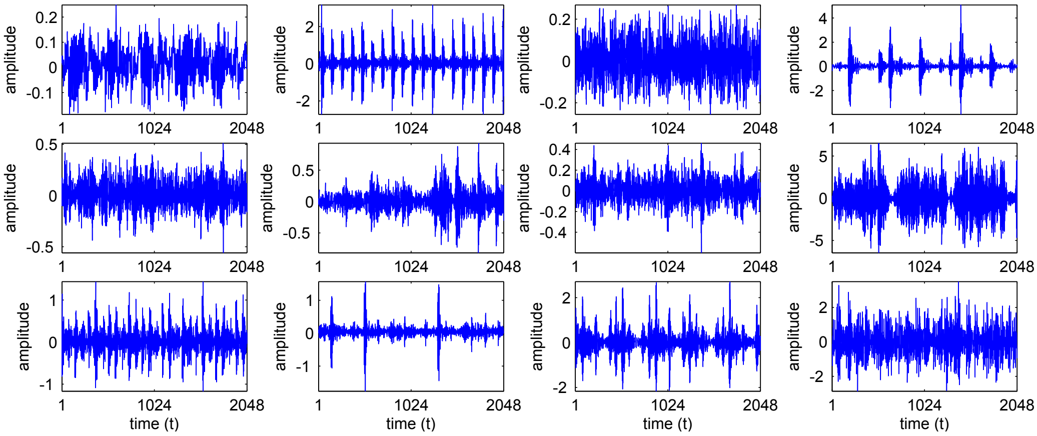


FIGURE 4.7: Time series for the twelve bearing conditions from the CWRU data set.

The first observation from Figure 4.8 is that SGDW1 algorithm is the best overall representation, reporting the best performance in 4 out of 5 distance measures evaluated and the second best in the remaining one (CO). The ACF representation is the second best overall technique and the best benchmark representation providing the lowest classification error, when using the Correlation distance; and reporting the second best result for the rest of the distance measures. The third best overall algorithm is the proposed SGDW3 which in 4 out of 5 distance measure provides the second best classification error. The above results suggest that vibration data from different motor conditions presents very specific autocorrelation patterns. This is the reason why not only ACF but also the autoregressive models i.e. ARIMA and ARMA are some the best representations for this kind of data. An additional observation from Figure 4.8 is that the structural generative strategy used by the proposed SGDW1 and SGDW3 algorithms, allow them to distinctively capture changes in the autocorrelation structure of data.

The second aspect that is important to highlight is that, as observed in Section 4.1.2.1

no matter which wavelet is selected the SGDW1 algorithm performs well. Specifically, in 3 out of 5 distance measures (ED, SE and CH) the classification error obtained using whichever wavelet W in the SGDW1 algorithm is lower than the error provided by any of the 21 benchmark algorithms studied. Contrary to the expectations, the SGDW2 algorithm, which was one of the top three algorithms in the previous experiment, reported a degraded performance when applied over vibration data, appearing in the group of the top ten representations only when the SE measure is used. Regarding SGDG-based algorithms, it can observed from Figure 4.8 that in 4 out of the 5 distances evaluated SGDG1 reports better results than SGDG2. The above results suggest that selecting coefficients as features, which is the case for SGDW1 and SGDW3, or selecting parameters as features, which is the case SGDG1, outperform strategies based on densities (i.e. SGDW2 and SGDG2). The reason for this is the fact that the corresponding coefficients or parameters, depending on the case, condense all the information contained in the density using a reduced number of features. Note that improved generalisation can be obtained with a reduced number of highly discriminative features instead of with a combination of discriminative, redundant and vague features. Moreover, the local approximation capabilities of the basis functions used in the density estimation stage of SGDW algorithms makes them superior than the SGDG algorithms that consider global Gaussian functions for the density estimation block. Here, it is important pointing out that even when SGDG1 is in general better than about the 75% of the evaluated benchmark representations, it is outperformed by ACF, ARIMA and ARMA algorithms when using ED, CO, CH or CR distance measures.

The pixel plots of Figure 4.9 are used to analyse the difference in performance in SGDW1 and SGDG1 algorithms when different wavelets W, different levels of decomposition M and different base resolutions j_0 are selected. Each of these pixel plots is related to one wavelet and is related to the averaged classification error over 100 trials using ED as distance measure. For ease of visualisation different colour scales are used for each method.

The main observation regarding Figure 4.9 is that, similar to what it was obtained for the Synthetic experiment, for both algorithms different wavelets provide good classification performance. Regarding the range of values for the parameters M and j_0 that provide low classification errors, it can be seen from Figure 4.9a and Figure 4.9b, that the SGDW1 algorithm provides good results with a wider range of values for its parameters than the SGDG1 algorithm. Note that this is related to the good localisation capabilities of the basis

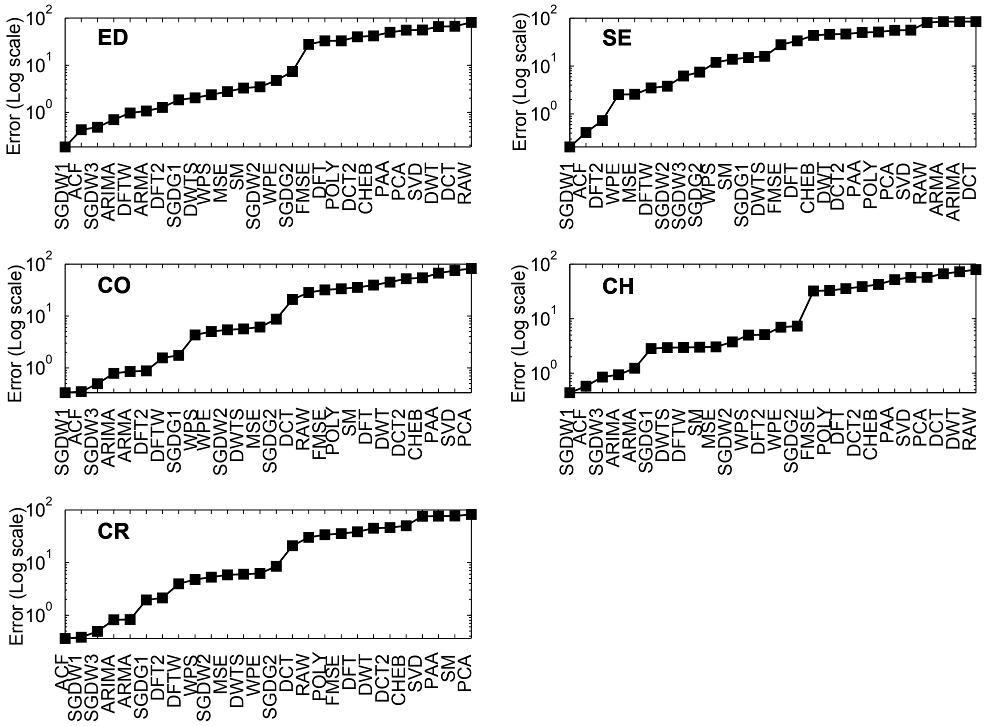


Figure 4.8: Errors for different distance measures for for CWRU experiment.

functions employed by density estimation technique considered in the SGDW1 algorithm. This implies that the resulting representations will be highly specific for different values of j_0. It is also important to highlight that, contrary to the result obtained with the synthetic data set of experiment of Section 4.1.2.2, in this experiment low classification errors are obtained when K, the number of Gaussians in the mixture of the density estimation stage, is equal to one. This result is directly related to the specific characteristics of the time series evaluated. Note that since the available vibration data is almost normally distributed then using a single Gaussian function would be enough to characterise the corresponding density. This also applies for the density of the decomposed subpatterns. The reason why more discrimination between classes is obtained using SGDs with $K = 1$ is related to the fact that the support of the additional Gaussian functions would be almost the same for time series of all classes, hence negatively impacting the generalisation performance of the classification algorithm.

4.1.2.3 ECG Biometrics Experiment

In the third experiment the proposed SGDs representations are evaluated in an ECG-based biometric application. Biometrics is concerned with the identification, verification

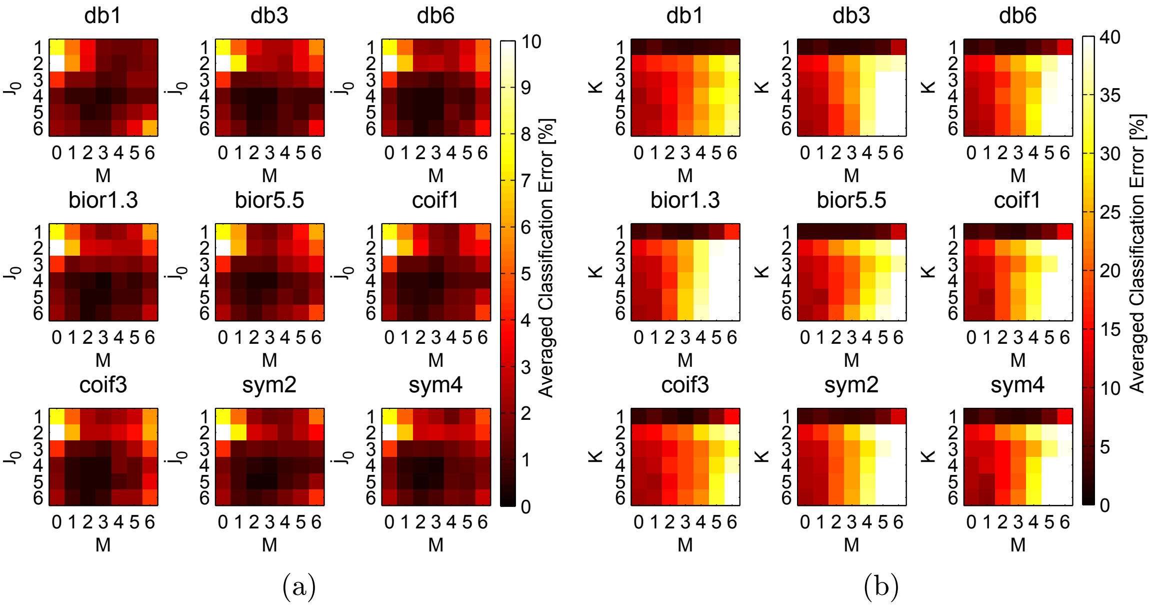


FIGURE 4.9: Pixel plots for SGDW1(a) and SGDG1(b) algorithms in CWRU experiment using ED as distance measure.

or screening of individuals based on their physiological (fingerprint, face recognition, palm print, hand and ear geometry, iris and retina scans) or behavioural (keystroke dynamics, gait analysis, signature and voice recognition) traits or characteristics [167, 168]. In recent years, there has been an increasing interest in the use of some particular biosignals, that is, electrocardiogram (ECG), electroencephalogram (EEG) and electrodermal response (EDR), for the design of more robust human identification systems. Among them, the ECG has attracted particular attention as a biometric trait since it not only have sufficient unique physiological properties to identify an individual but it also provides a real-time liveness feedback. ECG-based biometric techniques can be categorised into two main groups: *fiducial point dependent* techniques which rely on the detection of the so called fiducial points (such as P wave, QRS complex, T wave, etc.) and *fiducial point independent* approaches which are based on processing the ECG in a holistic manner without the need to localise fiducial points [169].

In this section the proposed SGDs representations are investigated in a fiducial point independent ECG biometric framework using the short-term rest ECG data base from [169–171]. This data base includes 3 minutes of one-lead ECG data from 48 individuals sampled at 200Hz extracted using the Vernier EKG-BTA sensor. The experiment considers the 16 individuals for which data from two different sessions is available. Specifically, excerpts of 9 seconds are extracted for each ECG recording to a create a data base of 640 time series of length 1800. The task consists in, using the 50% of the ECG excerpts from each individual for training, identify the individual from whom each of the remaining 50% of the ECG excerpts belong to. Figure 4.10 shows example ECG excerpts from the 16 individuals considered in the experiment.

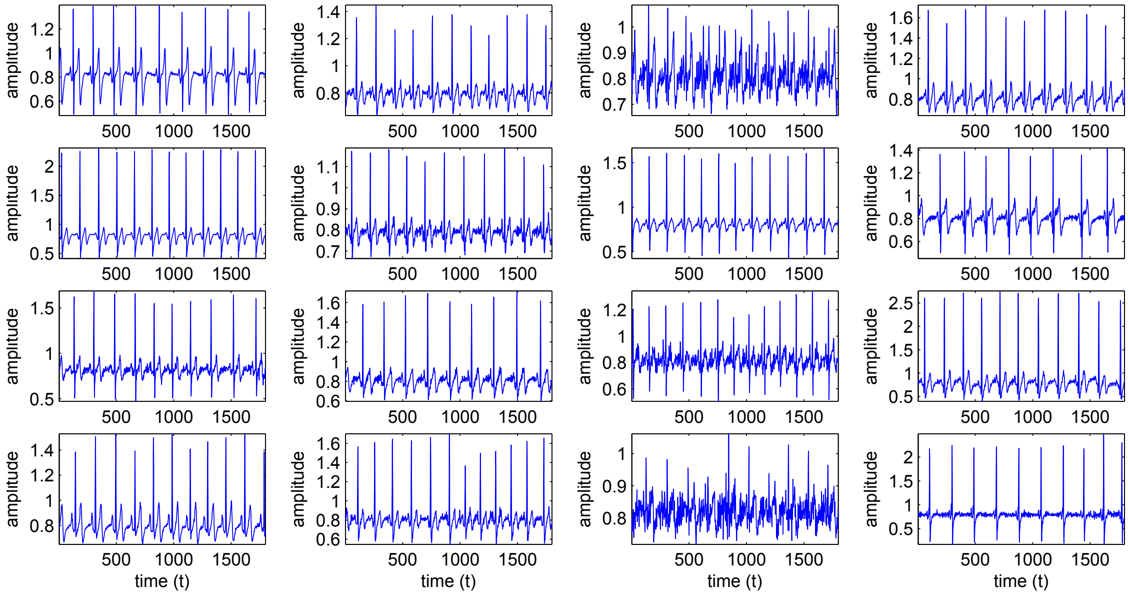


FIGURE 4.10: Example ECG excerpts from the 16 individual of the ECG data base.

Results and Discussions on the ECG Biometrics Data Experiment: In Figure 4.11 a summary of the performance results for this experiment are presented. For each distance measure, the time series representations are sorted according to the reported averaged classification error. For the case of SGDW-based and SGDG-based algorithms only the node related to the best wavelet is considered. Detailed results are presented in Table A.5 and Table A.6 from the Appendix.

The most important observation from Figure 4.11 is the fact that the proposed SGDW3 and SGDW1 algorithms are the best overall representations. They report the best performances for the five distance measures evaluated. The SGDW2 algorithm is the third best overall representation, providing the third best results in 4 out of 5 of distance measures. Regarding benchmark representations, ACF is the fourth best overall technique and the best benchmark representation. In respect to SGDG-based algorithms, the proposed SGDG2 and SGDG1 are the fifth and sixth best algorithms among all the representations evaluated.

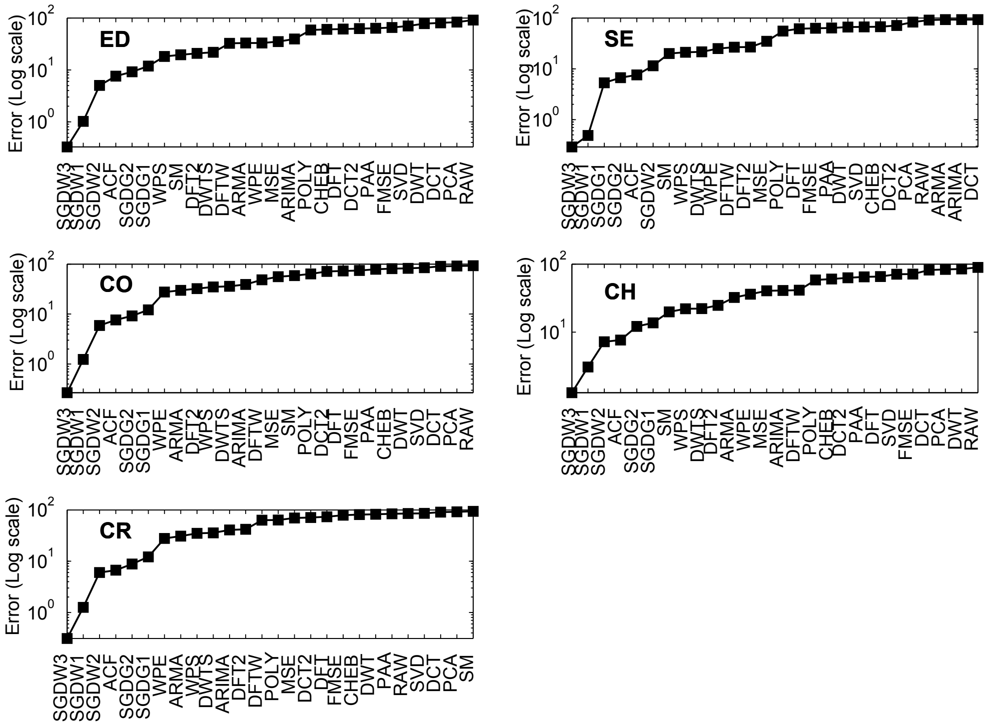
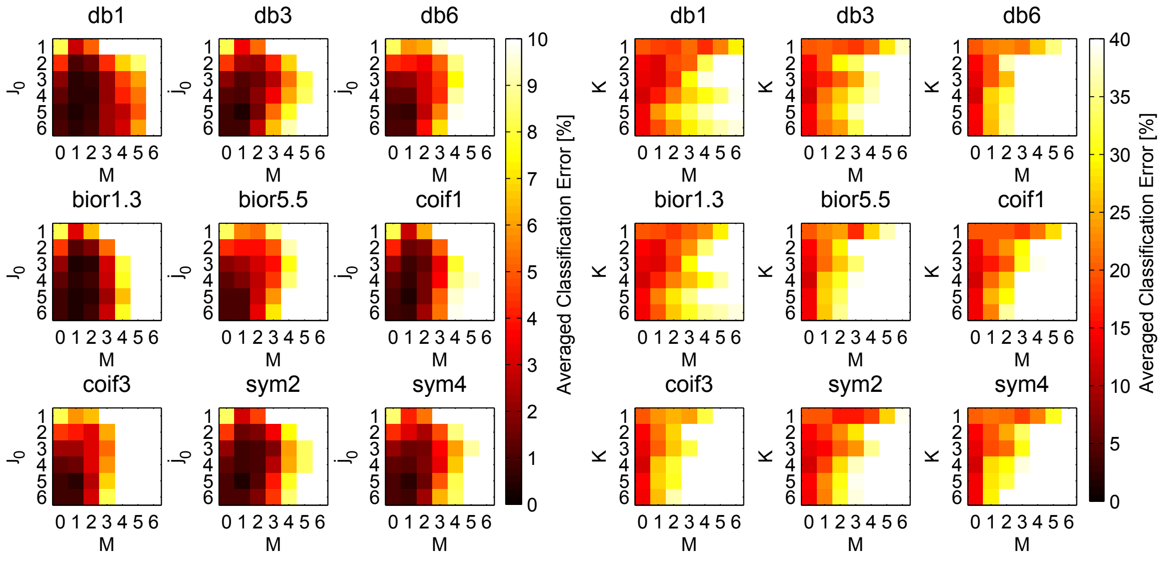


FIGURE 4.11: Sorted results for ECG biometrics experiment.

Another aspect that is important to highlight is that, similar to results obtained for Synthetic data and CWRU experiments, low classification errors in SGDW3 or SGDW1 algorithms can be obtained using whichever of the nine wavelets studied in the multiresolution decomposition stage. Note the fact that for the 5 distance measures evaluated the classification error obtained using whichever wavelet in SGDW3 and SGDW1 algorithms is lower than the error reported by any of the 21 benchmark representations evaluated. Note that the SGDW2 algorithm, which reported performances better than the ones obtained using any of the 21 benchmark representations in 4 out of 5 distance measures, offers favourable results only for some particular wavelets. Note also that regarding SGDG-based algorithms, the SGDG1 algorithm, which is the third best algorithm when using SE, also outperforms all benchmark representations with any of the nine wavelets for that distance measure.

The SGD-based algorithms are also assessed when different levels of decomposition M and different base resolutions j_0 or different numbers of Gaussian functions K (depending on the case) are used. Figure 4.12 shows pixel plots in which each pixel is related to the averaged classification error over 100 trials obtained with a particular combination of these parameters. Note that for each wavelet a different pixel plot is used and that for all the plots only ED is considered as distance measure. To make easier the visualization of results different colour scales for each method are used.

In Figure 4.12 can be observed that for both algorithms, different wavelets provide favorable classification performance. Moreover, similar to results obtained in the previous two experiments, there is also a specific range of values for the parameters M and j_0 or K that provide low classification errors. Specifically, as it can be seen from Figure 4.12a and Figure 4.12b, for SGDW3 the best results are obtained using $M \leq 2$ and $3 \leq j_0 \leq 6$ while for SGDG1 they are related to $M \leq 1$ for $2 \leq K \leq 6$ and $2 \leq M \leq 4$ for $j_0 = 1$. It is important to highlight here that, even though the range of parameters that provide a favourable performance is application specific, it is clear that this range will also depend on the modelling capabilities of algorithm selected for the density estimation stage. Results reported in Figure 4.12 indicate that if the number of probability domain subpatterns (related to parameter M) as well as their corresponding number of primitives (related to either parameter j_0 or parameter K, depending on the case) fall within a particular range then the resulting SGDs representations will provide an acceptable performance.



(a)

(b)

FIGURE 4.12: Pixel plots for SGDW3(a) and SGDG1(b) algorithms in ECG biometrics experiment using ED as distance measure.

4.1.2.4 Experiment with data from University of California Riverside

In the fourth experiment the proposed SGDs time series representations are evaluated using 42 benchmark data sets from the UCR time series clustering/classification repository [159]. Each data set includes time series of fixed length. The length of the time series varies from 60 to 1639 data items depending on the data set. The number of classes in the data sets goes from 2 to 50 classes and the number of time series per data set varies 56 from 9236. For each data set the experiment consists in using the 50% of the time series for training predict the corresponding class of the remaining 50%.

Table 4.5 shows the description of the UCR data sets, including the name of the data set, its abbreviation, the length of the corresponding time series, as well as the size of the data set (number of time series per data set) and the number of classes. An additional column is included indicating the maximum number of decomposition levels ($lmax$) for wavelet-based representations, namely, DWT, WPE, WPS, and DWTS, when $db6$ is selected as wavelet.

Results and Discussions on UCR Experiment: Results for this experiment, for both benchmark and proposed representations, are summarised in Table 4.6 and in Figure 4.13 to Figure 4.18. Table 4.6 shows the algorithm that reports the lowest averaged classification error for each of the 42 data sets studied and the five distance measure employed. Figure 4.13 presents the number of data sets for which a given representation reported the best result for each of the five distance measures. In addition to this, in Figure 4.14 to Figure 4.18 pixels plots are used to summarise the performance results for all time series representation in all the data sets evaluated. Each pixel plot corresponds to the classification performance using a particular distance measure. The axes of the plot are the representation algorithm (y-axis) and the data set (x-axis). In this way, each pixel represents the best averaged classification error obtained using a given algorithm in a particular data set. The colour of the pixel indicates the magnitude of the classification error. The darker the colour the smaller the classification error. The time series representations with the worst performance are located at the bottom of the plot. Note that pixel plots involves the arrangement of the data sets according to the average of the averaged classification errors obtained for all representations. Further details regarding this experiment are shown in Tables A.7 and Table A.8 from the Appendix A.

TABLE 4.5: Description of data sets.

n	**Data set**	**Abbr**	**Length**	**Size**	**Classes**	l_{max}
1	50words	50w	270	905	50	4
2	Adiac	Adc	176	781	37	4
3	Beef	Beef	470	60	5	5
4	CBF	CBF	128	930	3	3
5	ChlorineConcentration	ChlC	166	4307	3	3
6	CinC ECG torso	Cinc	1639	1420	4	7
7	Coffee	Coff	286	56	2	4
8	CricketX	CriX	300	780	12	4
9	CricketY	CriY	300	780	12	4
10	CricketZ	CriZ	300	780	12	4
11	DiatomSizeReduction	DiSR	345	322	4	4
12	ECG200	ECG2	96	200	2	3
13	ECGFiveDays	ECG5	136	884	2	3
14	FISH	Fish	463	350	7	5
15	FaceAll	FacA	131	2250	14	3
16	FaceFour	Fac4	350	112	4	4
17	FacesUCR	FacU	131	2250	14	3
18	Gun Point	GunP	150	200	2	3
19	Haptics	Hapt	1092	463	5	6
20	InlineSkate	InLS	1882	650	7	7
21	Lighting2	Ltg2	637	121	2	5
22	Lighting7	Ltg7	319	143	7	4
23	MALLAT	Mall	1024	2400	8	6
24	MedicalImages	MedI	99	1141	10	3
25	MoteStrain	MotS	84	1272	2	2
26	OSULeaf	OsuL	427	442	6	5
27	OliveOil	OliO	570	60	4	5
28	SonyAIBORobotSurfaceII	SnS2	65	980	2	2
29	SonyAIBORobotSurface	SnS	70	621	2	2
30	StarLightCurves	StLC	1024	9236	3	6
31	SwedishLeaf	SweL	128	1125	15	3
32	Symbols	Symb	398	1020	6	5
33	Trace	Trce	275	200	4	4
34	TwoLeadECG	2ECG	82	1162	2	2
35	Two Patterns	TWoP	128	5000	4	3
36	WordsSynonyms	WorS	270	905	25	4
37	Synthetic control	SynC	60	600	6	2
38	uWaveGestureLibrarX	uWGX	315	4478	8	4
39	uWaveGestureLibraryY	uWGY	315	4478	8	4
40	uWaveGestureLibraryZ	uWGZ	315	4478	8	4
41	Wafer	Wafr	152	7164	2	3
42	Yoga	Yoga	426	3300	2	5

The first observation from Table 4.6 and Figure 4.13 to Figure 4.18 is that, for the 42 data sets evaluated, there is not a particular time series representation that always brings the best performance. This finding agrees with Jain [172], in the sense that there is not a universally good data representation and, as it was recognised by [3], each representation generally tends to encode only those features well presented in its own representation space and inevitably incurs in the loss of useful information for the, in this case, classification tasks. These results demonstrate the difficulty in choosing an effective representation for a given time series data set without prior knowledge and careful analysis.

Table 4.6: Algorithm that reported the lowest averaged classification error over 100 trials for each data set in UCR Experiment.

Data set	**ED**	**SE**	**CO**	**CH**	**CR**
50w	PAA	PAA	PAA	CHEB	PAA
Adc	SGDW2	SGDW2	SGDW2	SGDW2	SGDW2
Beef	PCA	SVD	PCA	PCA	PCA
CBF	CHEB	PAA	CHEB	CHEB	CHEB
ChlC	SVD	SVD	PCA	PCA	SVD
Cinc	SVD	SVD	SVD	PCA	PCA
Coff	ACF	ACF	ACF	ACF	ACF
CriX	SGDW3	SGDW3	SGDW3	SGDW3	SGDW3
CriY	SGDW3	SGDW3	SGDW3	SGDW3	SGDW3
CriZ	SGDW3	SGDW3	SGDW3	SGDW3	SGDW3
DiSR	SGDW1	DWT	SGDW1	SGDW1	SGDW1
ECG2	SM	SM	SGDG1	SM	SGDG1
ECG5	DFT2	DFT2	DFT2	DFT2	DFT2
Fish	PCA	SVD	SVD	SVD	SVD
FacA	DCT2	DWT	CHEB	CHEB	CHEB
Fac4	PCA	PAA	PAA	DCT	PAA
FacU	DCT2	DWT	CHEB	CHEB	DCT2
GunP	DFT2	DFT	DFT2	DFT2	DFT2
Hapt	SGDW3	SGDG2	SGDW3	SGDW3	SGDW3
InLS	ARIMA	WPE	ARIMA	ARIMA	ARIMA
Ltg2	SGDW3	SGDG2	SGDG1	SGDW3	SGDW3
Ltg7	DCT2	PAA	DWT	DCT2	DWT
Mall	SGDW3	SVD	SGDW3	SGDW3	SGDW3
MedI	SGDW2	SGDW3	SGDW2	SGDW2	SGDW2
MotS	PAA	PAA	DWT	PAA	DWT
OsuL	SGDW3	SGDW3	SGDW3	SGDW3	SGDW3
OliO	SGDW1	SGDG2	SGDW1	SGDW1	SGDW1
SnS2	PAA	DCT2	DCT2	PAA	RAW
SnS	PAA	PAA	PCA	PCA	DCT2
StLC	SGDW3	SGDW3	SGDW3	SGDW3	SGDW3
SweL	SGDW2	DFT	SGDW2	SGDW2	SGDW2
Symb	SGDW3	SGDW3	SGDW3	SGDW2	SGDW3
Trce	POLY	POLY	POLY	POLY	WPE
2ECG	SVD	PCA	SVD	SVD	PCA
TWoP	DCT2	CHEB	PAA	DCT2	DCT2
WorS	PAA	PAA	DWT	CHEB	DWT
SynC	DCT2	DWT	SGDW3	CHEB	SGDW3
uWGX	PAA	PAA	DCT2	CHEB	DCT2
uWGY	DCT2	PAA	DCT2	PAA	DCT2
uWGZ	DCT2	PAA	DWT	DCT2	DWT
Wafr	DFT2	DFT2	DFT2	CHEB	DFT2
Yoga	DWT	DWT	DWT	PAA	DWT

Despite the above finding, it is important to highlight that the proposed SGDW3 algorithm offers a consistent performance in the experiments evaluated. According to results presented in Table 4.6 and Figure 4.13 this algorithm is the best overall representation, reporting the lowest classification error for a larger number of data sets than any other representation. Specifically, the SGDW3 algorithm is the best algorithm for ED, CO, CH, and CR distances, and the second best for SE.

In Table 4.6 and Figure 4.13, it can also be observed that, when the CO or CR are used as distance measures, in 16 out of 42 data sets the lowest classification error is obtained

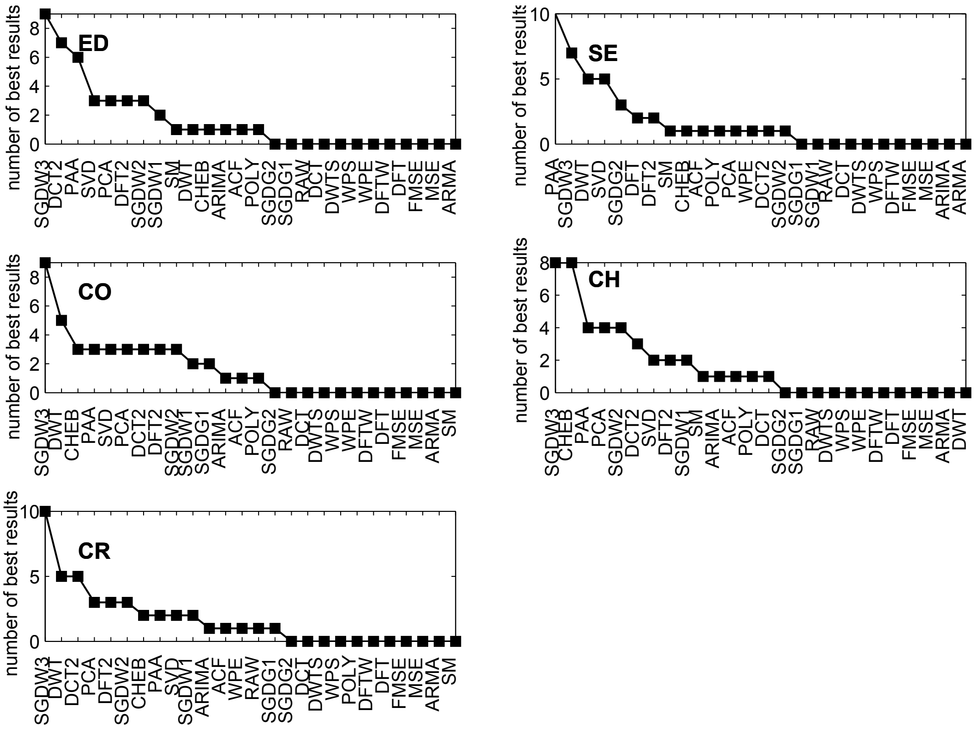


FIGURE 4.13: Number of data sets for which the representations evaluated reports the best result in the UCR experiment.

using one of the proposed algorithms. On the other hand, when ED or CH are considered in 14 out of 42 data sets, the proposed algorithms provides the best performance. For the case of SE SGD-based algorithms give the best result in 11 out of 42 data sets. An additional observation from Table 4.6 is the fact that for some particular data sets the proposed algorithms provide the best performance for the majority of the distance measures investigated. Here we can cite, Adc, CriX, CriY, CriZ, DiSR, Hapt, Lgt2. Mall, Medl, OsuL, OliO. StLC, SweL, Symb. Note also that many factors influence the classification complexity of a given data sets. Among them we could cite for instance, not only aspects related to the data sets themselves like the number of classes, the length and the complexity of the time series, but also characteristics intrinsic to the representation space of the time series representation algorithms selected (for example, the separation between classes and the geometrical complexity of class boundaries).

Special attention deserves PCA which appears on the top five representations according to Figure 4.14 to Figure 4.18. After a closer look into Table 4.6 and Figure 4.13, it can be can seen that although PCA reported competitive results for the majority of the data sets, it is the best algorithm in only 1 to 4 data sets, depending on the distance measure selected. The reason why PCA is consistently good in this context is related to the compression capabilities of this algorithm. Note that the highly dimensional nature of time series negatively impacts the generalisation performance of classification algorithms. Hence, PCA, which extracts a reduced number of features from data, enables classification algorithms to focus only on a few number of time series components with high variability and then, indirectly, it improves generalisation.

When SE is used as distance measure the proposed SGDW1 and SGDW3 algorithms show a degraded performance (see Figure 4.15). This is particulary more evident for DiSR, Mall, OliO, and Fish data sets. An explanation for is that the underlying normalisation procedure of this distance makes all the elements of the feature vector, related to the coefficients of the densities at different resolutions, to have the same standard deviation and as a consequence the same level of importance. This will make the elements of the feature vector related to regions of large difference in probability in the corresponding densities to be as important as the elements related to regions with small difference.

Regarding the pixel plots of Figure 4.14 to Figure 4.18, it can be seen that the proposed SGDs-based algorithms are located in the top half of the pixel plots. This means that the performance of all the five SGDs-based algorithms is at least better than the 50% of the benchmark representations evaluated in the experiment. A second observation is that according to the pixel plots corresponding to CO (Figure 4.16) and CR (Figure 4.18) the proposed SGDW3 is the best representation for this set of 42 benchmark time series. Moreover, when using ED (Figure 4.14) or CH (Figure 4.17) as distance measures, SGDW3 is the fourth best technique.

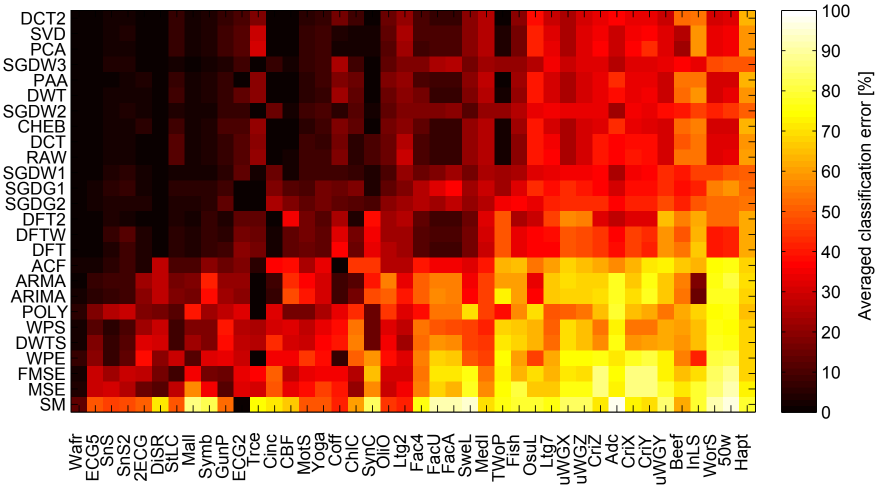


FIGURE 4.14: Results for UCR experiment using ED as distance measure.

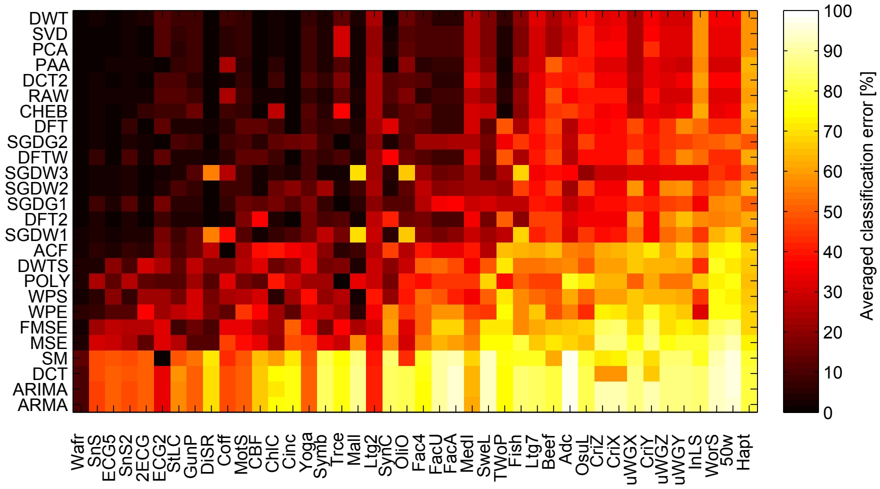


FIGURE 4.15: Results for UCR experiment using SE as distance measure.

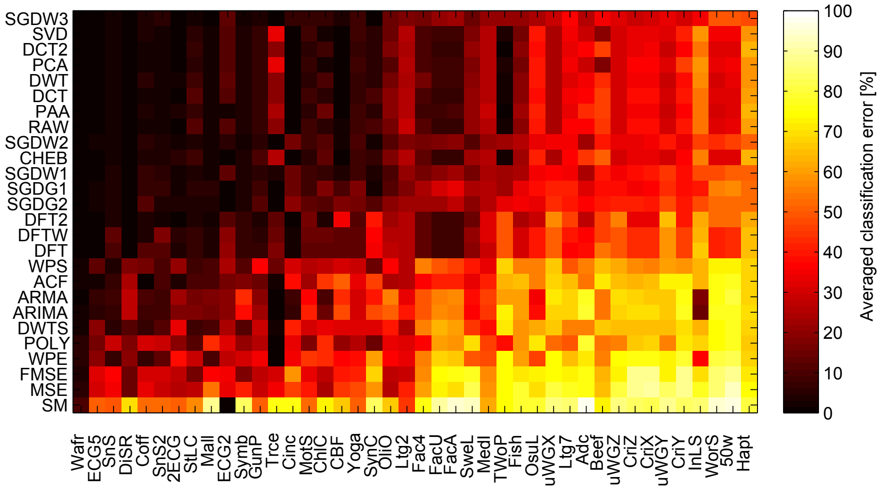


FIGURE 4.16: Results for UCR experiment using CO as distance measure.

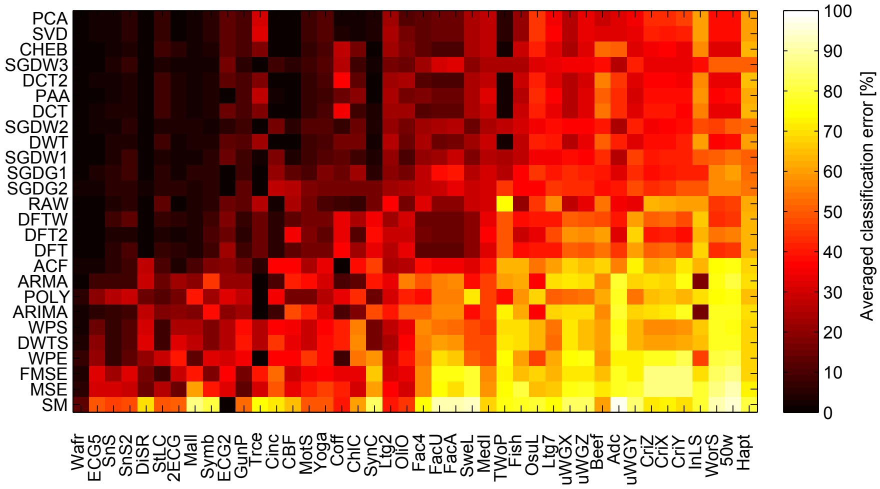


FIGURE 4.17: Results for UCR experiment using CH as distance measure.

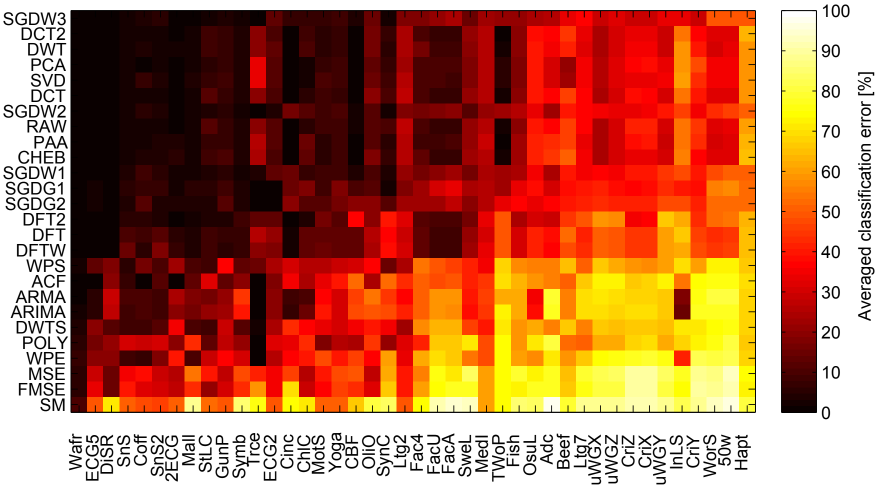


FIGURE 4.18: Results for UCR experiment using CR as distance measure.

4.2 Empirical Evaluation of Online SGDs Algorithms

The empirical evaluation of the proposed online SGDs algorithms is divided into two major parts. In the first experiment the proposed algorithms are assessed in terms of their usefulness in the context of change detection. The second set of experiments is focused on the applicability of the proposed algorithms in clustering of multiple parallel data streams. The assessment is carried out using synthetic and real world data.

4.2.1 OSGD-based Data Stream Change Detection Framework

Detecting changes in the properties of a given data stream is one of the key data stream mining tasks since it covers a broad range of related applications such as fault detection in engineering systems [173–176], intrusion detection in computer networks [177, 178], fraud detection in internet or online transactions [179, 180] and sensor networks [181, 182].

Change detection has been particularly studied by the statistics research community, where the traditional approach involves considering different probability densities of the data, each of them related to a different time interval. Usually two time intervals are used, the first one related to past values and the second one covering the most recent data available. In the statistical context, the change detection formulation relies on comparing the two densities to find if there is a significant difference between them. Relevant change point detection algorithms within this framework are CUSUM [13] and GLR [12]. It is important to highlight here, that the majority of the statistical-based change detection techniques are in essence parametric, since they rely on pre-specified autoregressive, state space or probability density models [10]. This dependency on a particular model is clearly the major limitation for parametric statistical change detection approaches.

The OSGDs algorithms proposed in Chapter 3 can be used to construct nonparametric data stream change detection algorithms that follow a formulation similar to the one above described for statistical-based techniques. However, since OSGDs data stream representations do not assume any particular model for the distribution of the data, the resulting approach overcomes the limitation regarding the pre-specification of any particular model. This is the reason why researchers have started to look for more flexible change detection solutions that do not necessarily rely on specific models, relevant works within this category are the methods reported in [11] and [10] which rely on the direct and relative estimation of the ratio of the probability densities, respectively.

In this section two different OSGDs representations are assessed, each of them associated with different weighting strategies for the RWDE stage. Note that a given weighting strategy in RWDE is related to the importance assigned to old or past data in respect to the new one. Hence, different weighting strategies will capture, indirectly, the temporal difference in the underlying generation process of the data. Therefore, the idea of the OSGD-based change detection algorithm here sketched is to use two OSGDs, one that gives more importance to old data and one focuses more on new data, and compare them using a distance measure. Note that, in order to allow the evaluation of the distance between the corresponding two OSGDs representations, they should share the same values for their corresponding parameters P and j_0 as well as they should use the same wavelets in both multiresolution decomposition and density estimation stages.

The block diagram for the proposed OSGD-based data stream change detection framework is depicted in Figure 4.19. In this figure the acronym OSDG1 is used to denote the online representation that puts more importance to new data while the term OSGD2 refers to the representation that focuses on old data instead. The notation w_1 and w_2, with $w_1 < w_2$, is used to denote the size of the window related to OSGD1 and OSGD2 respectively. Note that, as it was explained in Section 3.5, the discounting parameter θ that controls the number of data items or observations considered for the estimation of the density in the RWDE stage of the proposed algorithms is the inverse of the window size. Note also that the complexity of the proposed change detection framework can be reduced by considering that OSGD1 and OSGD2 share the same multiresolution stage and therefore only the RWDE stage is different. Figure 4.20 shows a modified version OSDG-based data stream change detection framework in which the same multiresolution stage is used for both OSGD1 and OSGD2 blocks.

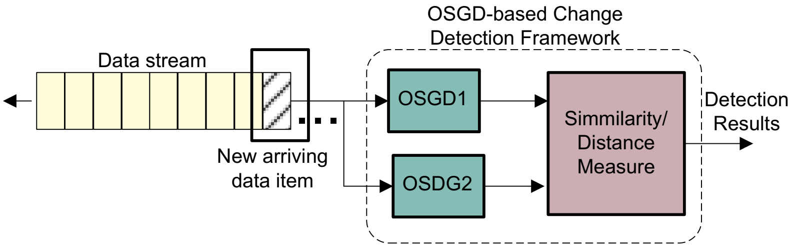


FIGURE 4.19: Block diagram for the proposed OSGD-based data stream change detection framework.

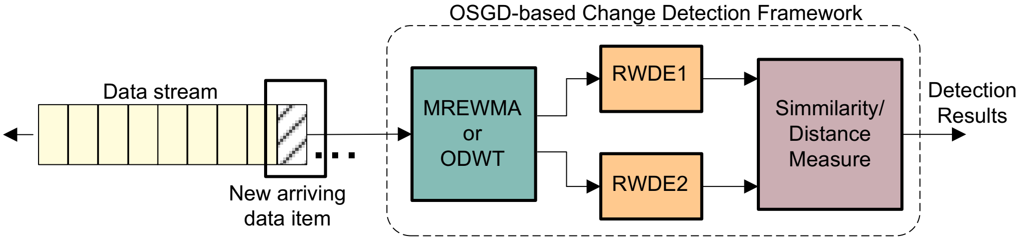


FIGURE 4.20: Block diagram for the modified OSGD-based data stream change detection framework.

Recall that, in the context of data streams, the change detection problem can be formulated in the following way. Let $\mathbf{x}^i = \{x^i(t)\}$ denote an ordered sequence of real valued observations taken at discrete times $t \in \mathcal{T} = \{1, 2, \ldots\}$, the objective of the change detection algorithm is to detect when changes in the properties of the underlying generation process of the data stream $\mathbf{x}^i$ have occurred.

4.2.1.1 Data Stream Change Detection Experimental Setting

The experiments in this section aim at assessing the potential usefulness of the proposed OSGDs algorithms in the context of change detection for data streams. For this purpose the proposed OSGD-based change detection framework is evaluated using synthetic data first. A data set of 240 streams, each of them containing 4000 data items, is constructed. Here, each stream belongs to one of the four main categories shown in Table 4.7, that is, change in mean, change in variance, change in frequency content and change in the deterministic generation process. Note that each category is in turn decomposed into three subcategories: slow change; moderate change; and abrupt change. There are in total 12 subcategories in the whole data set. The experiment consists in using the OSGD-based framework to detect the points in which the synthetic data streams change its underlying properties.

Table 4.7: Synthetic data set used for the change detection experiment.

Change	**Subtypes**	**Description**	**Time stamp of change**	**No. of Streams**
Change in mean	Slow linear change	From time stamps 1 to 2000 the data is drawn from $\mathcal{N}(0,1)$; from 2001 to 3000 the data of each time stamp ts is drawn from $\mathcal{N}(0.4*(ts-2000),1)$; and from 3000 to 4000 the data is drawn from $\mathcal{N}(4,1)$.	$2000-3000$	20
	Moderate linear change	From time stamps 1 to 2000 the data is drawn from $\mathcal{N}(0,1)$; From 2001 to 2500 the data of each time stamp ts is drawn from $\mathcal{N}(0.8*(ts-2000),1)$; and from 2501 to 4000 the data is drawn from $\mathcal{N}(4,1)$.	$2000-2500$	20
	Abrupt change	From time stamps 1 to 2000 the data is drawn from $\mathcal{N}(0,1)$ while from 2001 to 4000 the data is drawn from $\mathcal{N}(4,1)$.	At 2000	20
Change in variance	Slow linear change	From time stamps 1 to 2000 the data is drawn from $\mathcal{N}(0,1)$; from 2001 to 3000 the data of each time stamp ts is drawn from $\mathcal{N}(0,1+0.004*(ts-2000))$; and from 3000 to 4000 the data is drawn from $\mathcal{N}(0,3)$.	$2000-3000$	
	Moderate linear change	From time stamps 1 to 2000 the data is drawn from $\mathcal{N}(0,1)$; From 2001 to 2500 the data of each time stamp ts is drawn from $\mathcal{N}(0,1+0.008*(ts-2000))$; and from 2501 to 4000 the data is drawn from $\mathcal{N}(0,3)$.	$2000-2500$	20
	Abrupt change	From time stamps 1 to 2000 the data is drawn from $\mathcal{N}(0,1)$ while from 2001 to 4000 the data is drawn from $\mathcal{N}(0,3)$.	At 2000	20
Change in freq. content	Slow linear change	From time stamps 1 to 2000 the data is generated from $sin(0.07*ts)$; from 2001 to 3000 the data of each time stamp ts is generated from $sin(0.07*ts)+\mathcal{N}(0,9^{-5}*(ts-2000))$; and from 3000 to 4000 the data is generated from $sin(0.07*ts)+\mathcal{N}(0,0.09)$.	$2000-3000$	20
	Moderate linear change	From time stamps 1 to 2000 the data is generated from $sin(0.07*ts)$; from 2001 to 2500 the data of each time stamp ts is generated from $sin(0.07*ts)+\mathcal{N}(0,1.8^{-4}*(ts-2000))$; and from 2501 to 4000 the data is generated from $sin(0.07*ts)+\mathcal{N}(0,0.09)$.	$2000-2500$	20
	Abrupt change	From time stamps 1 to 2000 the data is generated from $sin(0.07*ts)$ while from 2001 to 4000 the data is generated from $sin(0.07*ts)+\mathcal{N}(0,0.09)$.	At 2000	20
Change in deterministic generation process	Slow linear change	From time stamps 1 to 2000 the data is generated from $0.7*sin(0.06*ts)+0.3*sin(0.18*ts)$; from 2001 to 3000 the data of each time stamp ts is generated from $(0.7-2^{-4}*ts)*sin(0.06*ts)+(0.3+2^{-4}*ts)*sin(0.18*ts)$; and from 3000 to 4000 the data is generated from $0.5*sin(0.06*ts)+0.3*sin(0.18*ts)$.	$2000-3000$	20
	Moderate linear change	From time stamps 1 to 2000 the data is generated from $0.7*sin(0.06*ts)+0.3*sin(0.18*ts)$; from 2001 to 2500 the data of each time stamp ts is generated from $(0.7-4^{-4}*ts)*sin(0.06*ts)+(0.3+4^{-4}*ts)*sin(0.18*ts)$; and from 2501 to 4000 the data is generated from $0.5*sin(0.06*ts)+0.5*sin(0.18*ts)$.	$2000-2500$	20
	Abrupt change	From time stamps 1 to 2000 the data is generated from $0.7*sin(0.06*ts)+0.3*sin(0.18*ts)$ while from 2001 to 4000 the data is generated from $0.5*sin(0.06*ts)+0.5*sin(0.18*ts)$.	At 2000	20

The performance of the proposed algorithms is evaluated using different pairs of window sizes w_1 and w_2 as well as different values for parameters M and j_0. The Euclidean distance is used to assess the dissimilarity between the two OSGDs instantiations employed in the change detection frameworks. Note that in order to compare the two OSGDs instantiations considered by the framework the corresponding representations provided by the two OSGDs algorithms should comprise the same number of elements and for that reason their parameters M and j_0 should be set to be the same.

Figure 4.21 and Figure 4.22 show an example data set from each of the 12 subcategories present in the data set as well as the their corresponding OSGD-E-based change detection result for different pairs of window sizes. For these figures, the OSGDs algorithm selected use one level of decomposition in the ODWT or MREWMA stage ($M = 1$) and a base resolution equal to 2^{-2} in the RWDE stage ($j_0 = 2$). In Figure 4.21 and Figure 4.22 the colour axis, whose scale is the same for all subfigures, represents the Euclidean distance between OSGD1 and OSGD2. In this way, the darker the colour larger the dissimilarity between the OSGDs representation using w_1 as window size respect to the one using w_2. Note also that for a better visualisation, only the last 3000 data items are plotted.

The first observation from Figure 4.21 and Figure 4.22 is that different pairs of window sizes bring different detection capabilities in terms of the Euclidean distance between the two OSGDs representations employed. The second observation is that such detection capabilities depend on the nature of the change present in the data. In this respect, the streams for which larger Euclidean distance were obtained after a given change in the stream are the ones involving changes in the mean of the underlying generation process of the data. On the other hand, the streams from the four category, which involves changes in the deterministic properties of the data, are the most difficult detection scenarios since only for a reduced number of pair of windows the Euclidean distance reported a significant increment.

By considering that the change in the properties of the data streams studied is set to start at time stamp 2001, then it can be seen that, as it is theoretically expected, there is a shorter delay in the detection of data streams with abrupt changes (streams 3 and 6 in Figure 4.21 and streams 9 and 12 in Figure 4.22). This means that there is a bigger difference between OSGD1 and OSGD2 for such cases, which in turn indicates that the proposed data stream representations properly characterised both old and new data.

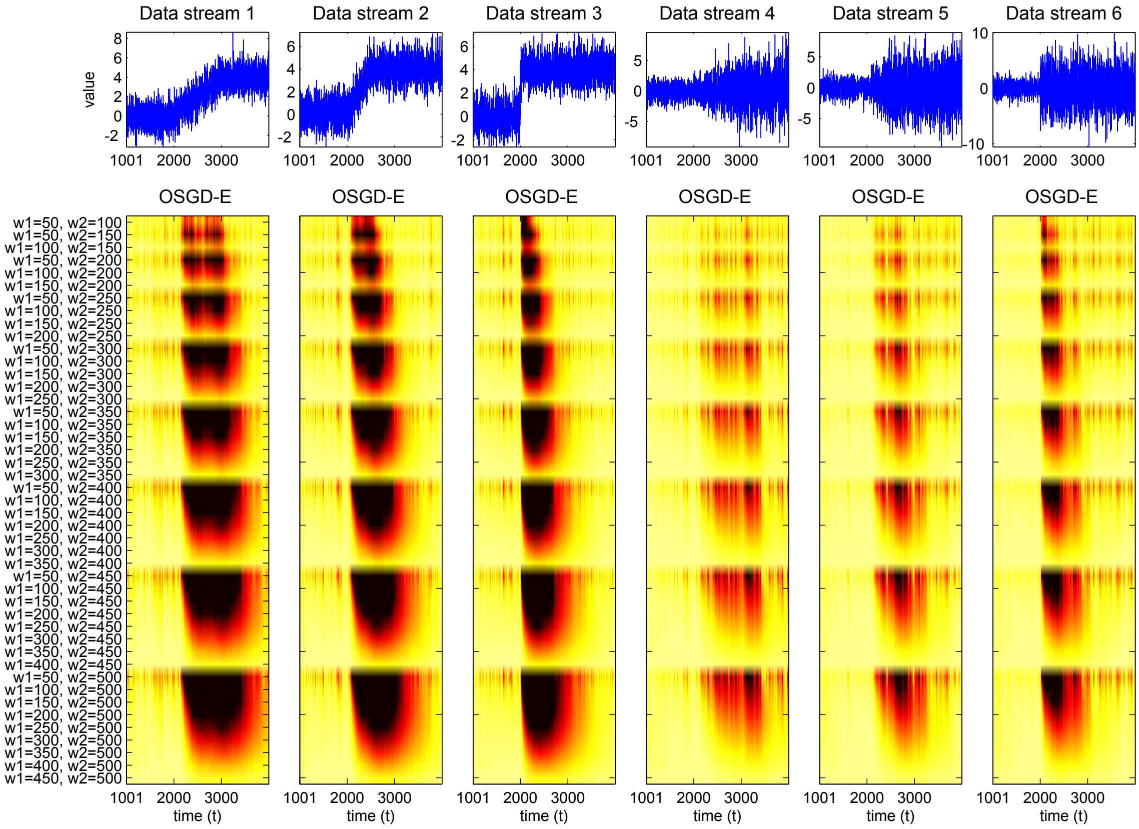


FIGURE 4.21: Example data streams from the first six subcategories of the data set and their corresponding Euclidean distance obtained using the proposed change detection framework using OSGD-E representations.

It is clear that the optimum pair of window sizes will depend on the nature of the changes required to be detected and, in that sense, one of the key strengths of the algorithm is that the size of the windows involved, which is related to the horizon for the analysis, can be modified accordingly without increasing the complexity of the algorithms. In this way, the OSGD-based change detection framework offers the possibility of choosing between a wide range of window size combinations. For instance, for the case of data stream 1 in Figure 4.21 window sizes that would guarantee good detection results would be all those pair of windows whose difference is larger than 200 ($w_2 - w_1 > 200$) with the size for the first window set to be $w_1 < 250$. Note however, that reducing the window size in OSGDs algorithms implies directly reducing the number of data items involved in the density estimation stage and, in that sense, constructing probability domain subpatterns that ignore most of the past values of the stream.

Note also that although changes related to streams from the four main category (changes in the deterministic properties of the stream) are more difficult to detect, an appropriate

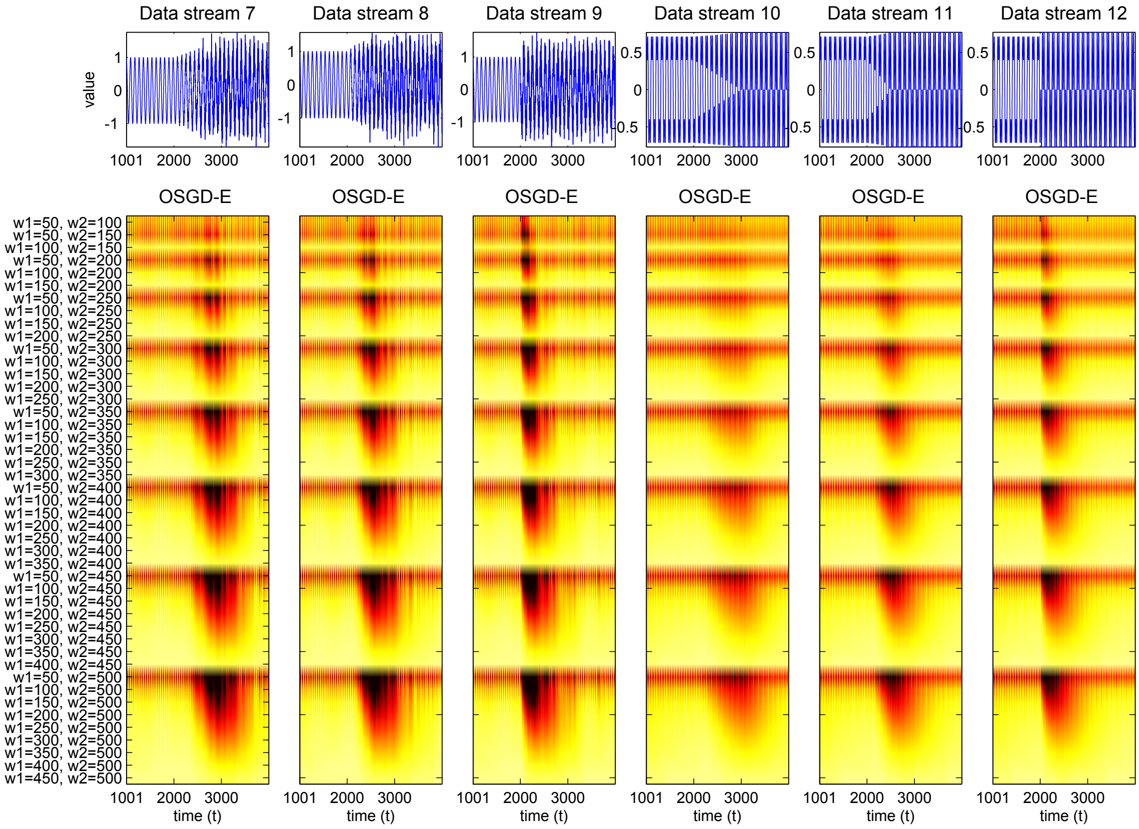


FIGURE 4.22: Example data streams from the first six subcategories of the data set and their corresponding Euclidean distance obtained using the proposed change detection framework using OSGD-E representations.

pair of window sizes for the proposed OSGD-based change detection framework still can be found. In the reminder of this section results regarding OSGD-D representations are omitted since they are very similar to the ones presented for OSGD-E in Figure 4.21 and Figure 4.22.

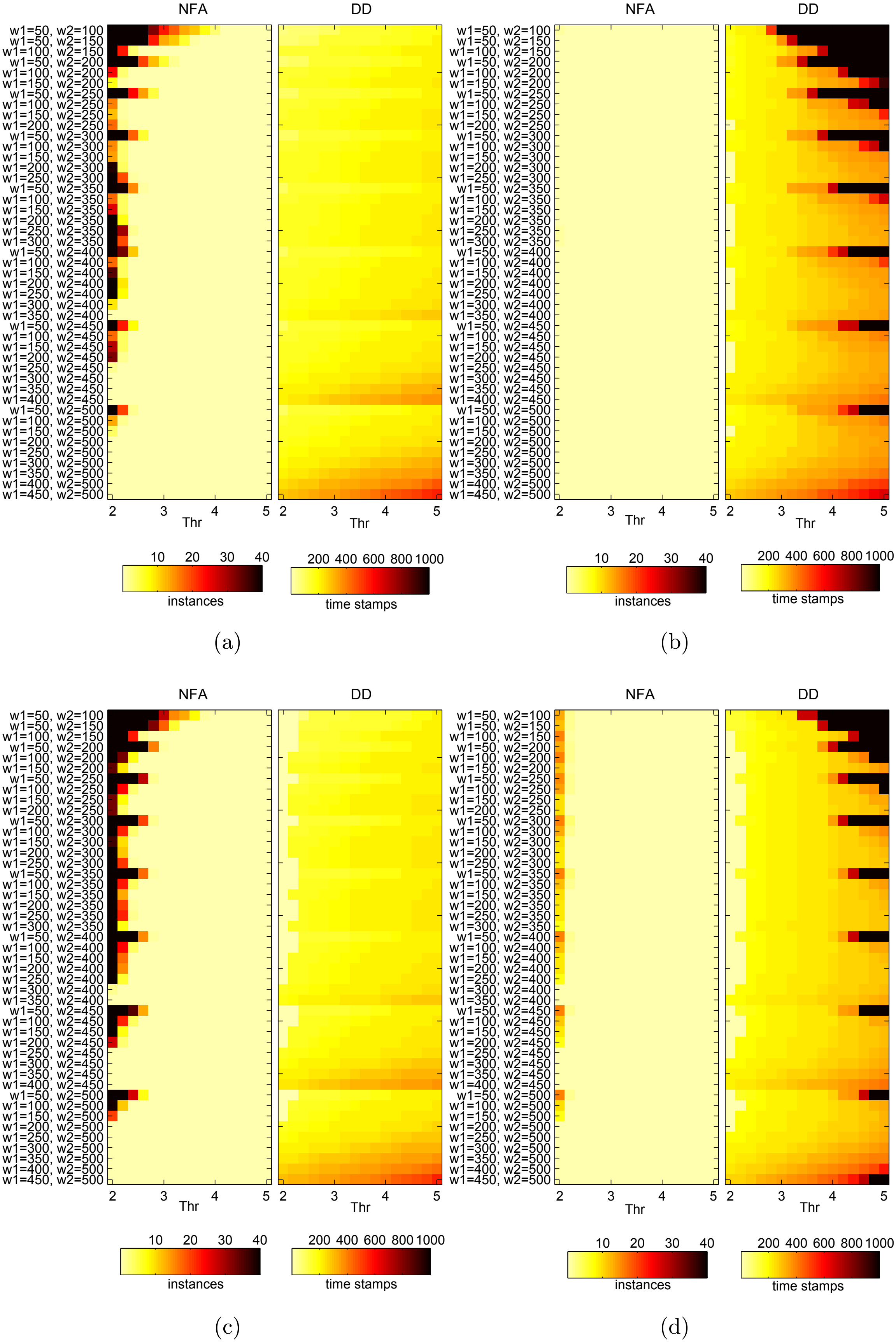
The performance of the OSGDs framework is investigated in terms of the number of false alarms (NFA) detected as well as in terms of the detection delay (DD) incurred when identifying changes. For this purpose, a simple threshold *thr* that is applied over the resulting Euclidean distance computed between OSGD1 and OSGD2 is defined. This threshold accounts for a specified number of times that the mean of the Euclidean distances when there is no change involved is exceeded. For instance, when *thr* is set to 2 the detected changes in the stream will refer to all those situations in which the mean of the Euclidean distance between OSGD1 and OSGD2 is higher than two times the mean of the distance associated to no change.

For the twelve subcategories of streams in the synthetic data set the mean of the Euclidean distances when there is no change involved are obtained by averaging the distances associated with time stamps 1000 to 1050 in each stream. Then, the resulting means corresponding to streams from the same subcategory are averaged.

Once the threshold thr has been defined, the NFA and the DD can be obtained considering the time stamps and duration of the changes present in each stream from Table 4.7. For each of the 240 streams, the NFA as well as the DD are calculated using as reference the true changes identified when different pairs of window sizes are selected in OSGD-based detection framework. For visualisation purposes pixel plots are constructed in which each pixels is associated with either the averaged detection delay or the averaged number of false alarms, depending on the case, obtained for the 20 streams from the same subcategory using a particular pair of window sizes as well as specific values for parameters M and j_0.

Figure 4.23 and Figure 4.24 show the performance of the OSGD-based change detection framework when different values for the parameter M (in OSGD-D algorithm) or parameter j_0 (in OSGD-E algorithm) are selected. Note that these figures are related to data streams with the first type of change. Also note that the same scale is employed for each pixel plot related to the same metric. The main observation from Figure 4.23 and Figure 4.24 is that, as it expected for change detection algorithms, there is a tradeoff between NFA and DD and, in that sense, for each pair of windows, as NFA decreases the corresponding DD increases and viceversa. The second observation is that in general, OSGD-E presents a reduced NFA while maintaining competitive results for DD in respect to results reported using OSGD-D. Note that, for all combinations of parameters, except for $M = 2$, $j_0 = 3$ (Figure 4.24d), the NFA reported by the OSGD-E algorithm is smaller than the one obtained using OSGD-D with the same parameters. Another aspect that is worth to highlight from Figure 4.23 and Figure 4.24 is that regarding the parameter j_0, the best results in terms of both NFA and DD for both algorithms are obtained when $j_0 = 3$, except for Figure 4.24d. Note that j_0 is the parameter related to the base resolution in RWDE and, in that sense, it controls the localisation capabilities of the resulting density estimate. Density estimates obtained using $j_0 < 3$ are estimates that capture the main trends in the density but are not suitable for representing discontinuities and local oscillations.

Figure 4.25 and Figure 4.26 show the averaged NFA and the averaged DD for the twelve subcategories shown in Table 4.7 using OSGD-D as the method for the change detection



(a)

(b)

(c)

(d)

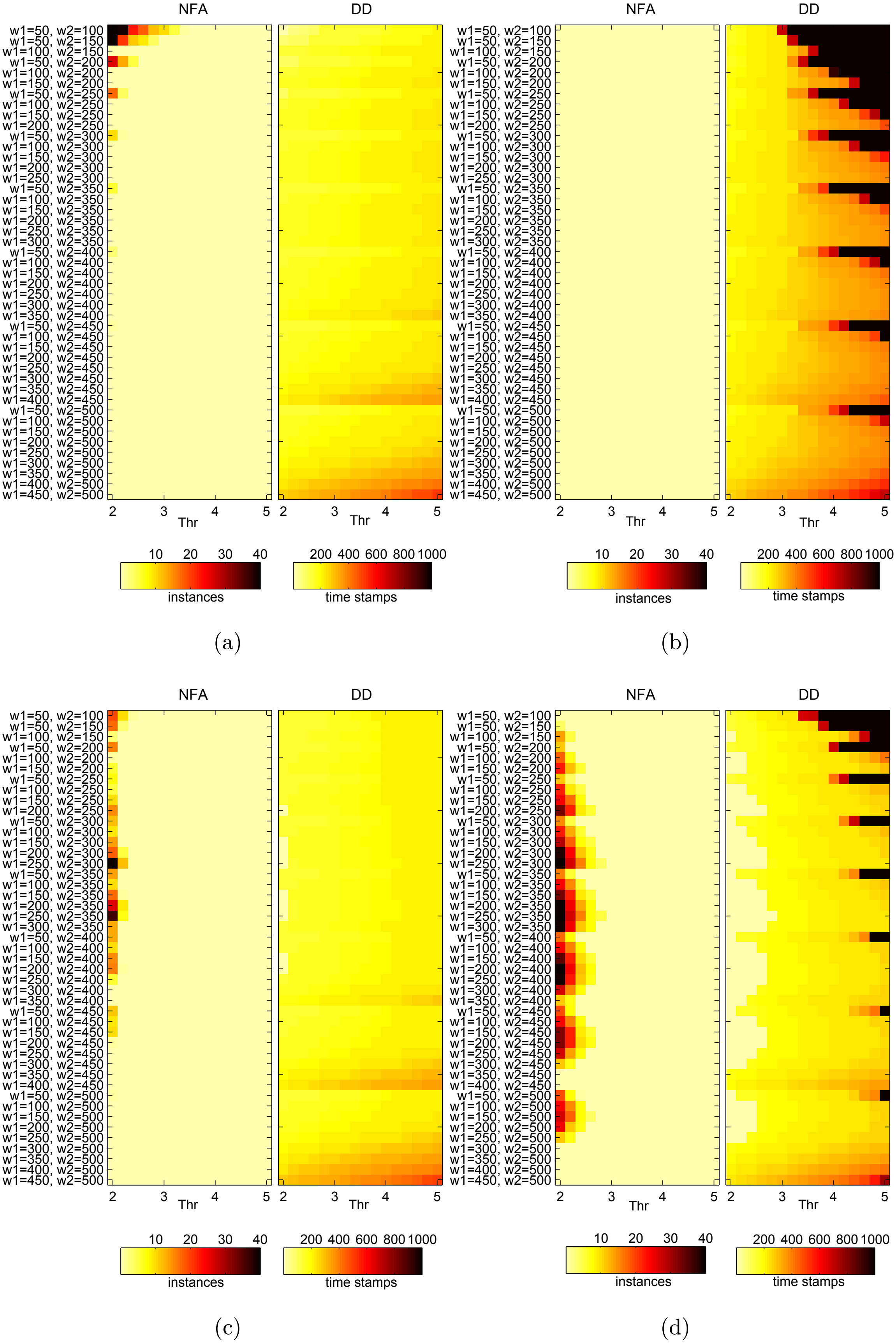
FIGURE 4.23: Pixel plots for NFA and DD in OSGD-D-based detection algorithm for different pair of window sizes and different values of the threshold *thr*: (a) $M = 1$ and $j_0 = 2$; (b) $M = 1$ and $j_0 = 3$; (c) $M = 2$ and $j_0 = 2$, (d) $M = 2$ and $j_0 = 3$.

framework. For these figures the number of decompositions levels M is set to 1, while the base resolution j_0 is set to 2. Figure 4.25 shows results for the first four subcategories of changes: (a) Slow linear change in mean, (b) Moderate linear change in mean, (c) Abrupt change in mean, (d) Slow linear change in variance. Figure 4.26 shows results for the fifth to the ninth type of changes: (a) Moderate linear change in variance, and (b) Abrupt change in variance, (c) Slow change in frequency content, (d) Moderate change frequency content. Finally, Figure 4.27 depicts results for the last four subcategories: (a) Abrupt change in frequency content, (b) Slow change in deterministic properties, (c) Moderate change deterministic properties, and (d) Abrupt change in deterministic properties. Note that in Figures 4.25 and Figure 4.26 the black colour in DD pixel plots indicates that the corresponding algorithm fail to detect changes within the 1000 data items posterior to the starting point of the change. On the contrary, non black pixels in DD pixel plots, means that the pair of window sizes are able to detect the underlying changes in the properties of the data stream. Regarding NFA pixel plots, white pixels means that there are no false alarms detected in such cases, while black pixels are associated with situations in which the number of false alarms is equal or higher than 40.

As it can be observed from Figure 4.25 to Figure 4.27 the most difficult type of change to be detected is the one related to changes in the underlying deterministic generation process of the data (Figure 4.27b to Figure 4.27d). Furthermore, among the 12 subcategories studied, slow changes in deterministic properties (Figure 4.27b) is the most challenging kind of change. The reason for this is that the particular type of deterministic change studied here involves a data stream that keeps its stochastic properties (mean and variance) almost unaltered. In this way, changes can only be detected by capturing subtle differences in the probability density of the data. Note here that the proposed OSGDs representations are able to capture such differences since they are based on estimating the probability density of the data at different resolutions.

4.2.2 Clustering of Multiple Data streams

One of the fundamental mechanisms for understanding and learning is the organisation of data into meaningful or natural groups [172]. Cluster analysis is the formal study of methods and algorithms for grouping objects according to their intrinsic characteristics or similarity [172]. Clustering aims at identifying the underlying structure in an unlabeled



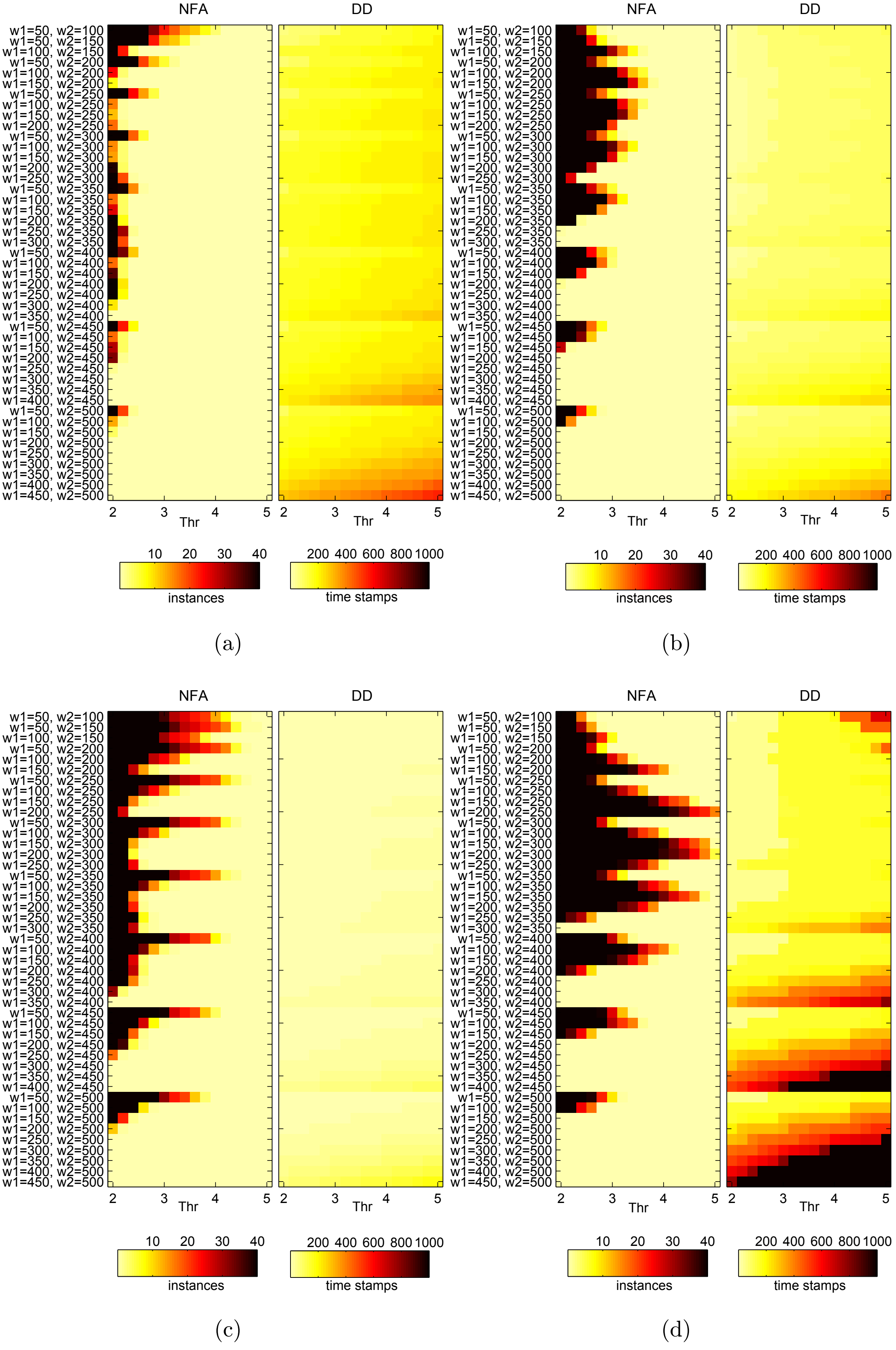
(a)

(b)

(c)

(d)

FIGURE 4.24: Pixel plots for NFA and DD in OSGD-E-based detection algorithm for different pair of window sizes and different values of the threshold *thr*: (a) $M = 1$ and $j_0 = 2$; (b) $M = 1$ and $j_0 = 3$; (c) $M = 2$ and $j_0 = 2$, (d) $M = 2$ and $j_0 = 3$.



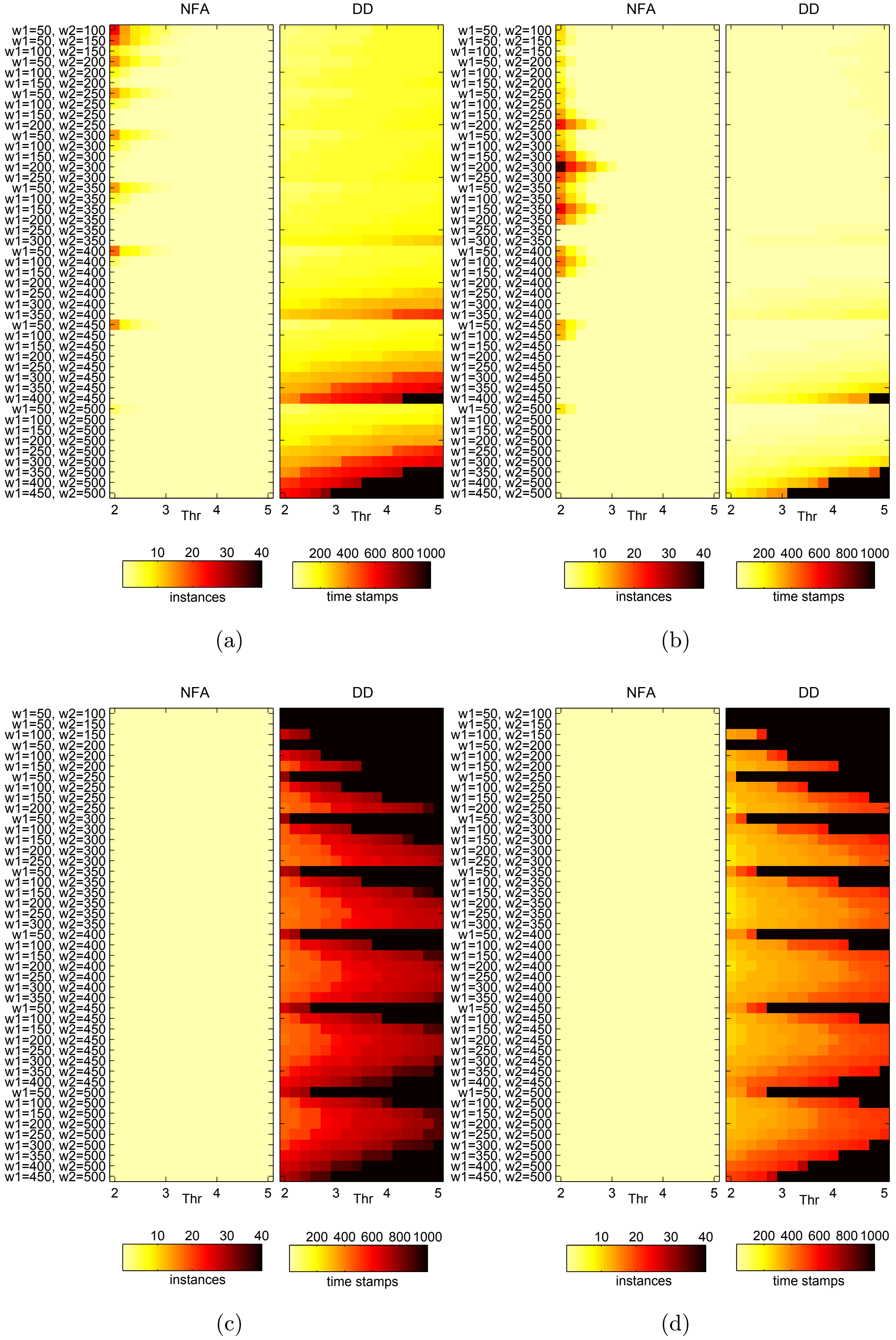
(a)

(b)

(c)

(d)

Figure 4.25: Pixel plots for NFA and DD using OSGD-D-based detection algorithm for the first four subcategories of changes studied using $M = 1$ and $j_0 = 2$.



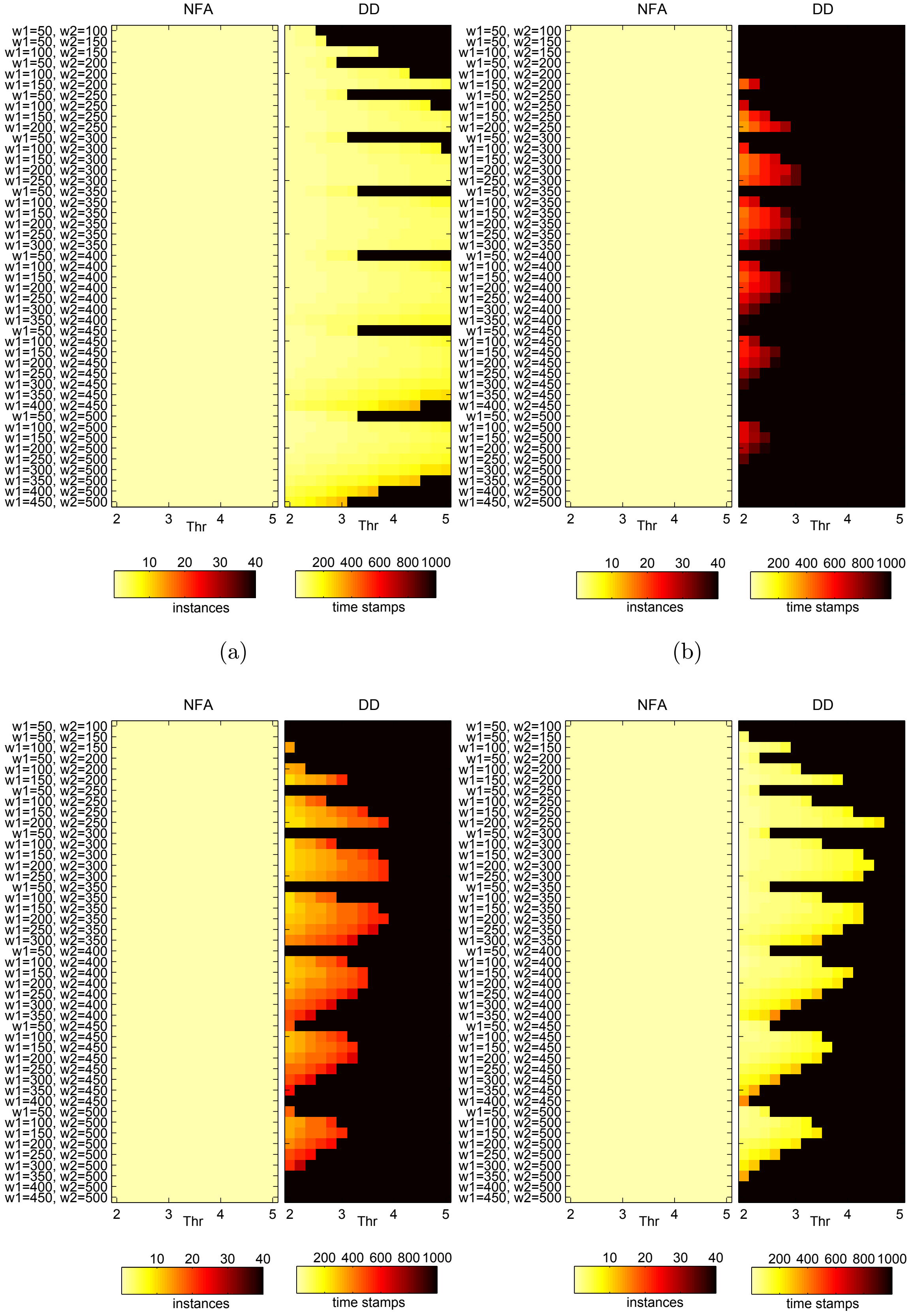
(a)

(b)

(c)

(d)

FIGURE 4.26: Pixel plots for NFA and DD using OSGD-D-based detection algorithm for the fifth to ninth subcategory of changes studied using $M = 1$ and $j_0 = 2$.



(a)

(b)

(c)

(d)

Figure 4.27: Pixel plots for NFA and DD using OSGD-D-based detection algorithm for the last four subcategories of changes studied using $M = 1$ and $j_0 = 2$.
]]>

data set by systematically organising data into homogeneous groups, in such a way that the similarity among data items within the same group and the dissimilarity among data items from different groups are jointly maximised [2].

In general, as it was highlighted in Section 3.2.3, in data stream mining tasks, such as clustering, classification, rule discovery, segmentation and summarisation, a fundamental issue that must be addressed is finding an appropriate representation for streaming data [81]. Two are the main reasons that make data representation essential for clustering tasks: 1) reduction in dimensionality [81], and 2) improvement in cluster stability [172]. Specifically, in clustering applications, data representation is a crucial factor that directly influence the performance of the algorithms due to the fact that a good data representation will produce compact and isolated clusters that can be easily identified even by a simple clustering algorithm [172].

In this section an online SGDs-based clustering framework is proposed. In the proposed framework the OSGDs algorithms proposed in Section 3.5 are used to extract online representations for streaming data. Similar to the framework reported in [103, 106] and [183], the proposed data stream clustering framework divides up the clustering process into online and offline phases. In the online phase, incremental representations of the data streams are constructed using the proposed OSGDs-based algorithms. In the offline phase the corresponding clustering of such representations is performed using an incremental clustering technique. The block diagram for the proposed OGSDs-based data streams clustering framework is depicted in Figure 4.28.

Since the proposed OSGDs representations are expressed as fixed-length vectors, their subsequent clustering, at each time stamp t, can be performed using any feature-based technique. As a proof of concept, an online implementation of the popular k-means algorithm is proposed as the clustering algorithm. Note that in [103] k-means was adapted for its application in online contexts, suggesting an incremental strategy that considers the cluster centers obtained at time stamp t as the initial values for centres at the subsequent time stamp $t+1$. For the sake of completeness, its theoretical foundations are briefly reviewed.

> **The k-means algorithm**: The k-means algorithm is the most popular and the simplest partitional clustering method. The goal of k-means is to find the best partition of a data set $\mathbf{X}$ into K groups such that the squared error between the

cluster's centroid and the data points included in a given cluster is minimised [172]. Specifically, this algorithm tries to minimise the sum of the squared error over all clusters expressed by the following objective function:

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \parallel \mathbf{x}_i - \mathbf{m}_k \parallel^2 \tag{4.2}$$

where $\parallel \mathbf{x}_i - \mathbf{m}_k \parallel^2$ is the squared Euclidean distance between a data point $\mathbf{x}_i$ and the cluster's centroid $\mathbf{m}_k$. More details about the formulation and extensions of this algorithm can be found in [172].

Note that what makes conceptually different the proposed SGDs-based clustering framework respect to solutions reported in [103, 106] and [183], is the fact that the proposed approach makes use of more robust features that combine structural and statistical aspects without compromising the resulting computational complexity.

The problem of clustering multiple parallel data streams assumes a set of N univariate data streams $\mathbf{X} = \{\mathbf{x}^i\}, i \in \mathcal{N} = \{1, 2, \ldots, N\}$, here each $\mathbf{x}^i = \{x^i(t)\}$ is an ordered sequence of real valued observations taken at discrete times $t \in \mathcal{T} = \{1, 2, \ldots\}$. The objective of the clustering algorithm is to seek, at time stamp t, a K-partition of $\mathbf{X}$, $\mathbf{C} = \{C_1, C_2, \ldots, C_K\}$ with $K \leq N$, such that $C_i \neq \emptyset$; $\cup_{i=1}^{K} C_i = \mathbf{X}$ and $C_i \cap C_j = \emptyset$ for all $i, j = 1, \ldots, K$ with $i \neq j$.

4.2.2.1 Data Stream Clustering Experimental Setting

The data used for the evaluation is the daily 5 km x 5 km mean temperature data set for the UK from the UK's National Weather Service (Met Office)[184], which includes $10,359$ cells, identified using the Ordnance Survey National Grid. The grid covers the whole of the UK (including Northern Ireland). Each cell consists of $17,167$ data items, where each item represents an estimate for the mean temperature of the centre point of the 5 x 5 km grid cell for a given day between 1960-2006. In the evaluation each time series of mean temperatures for a given cell is treated as a single data stream and hence, the data set includes $10,359$ streams.

The experiment consists in obtaining, at every time stamp, OSGDs representations (using either OSGD-D or OSGD-E algorithms) for the $10,359$ data streams and then cluster the resulting feature vectors using incremental k-means [103]. In the evaluation a pre-defined number of clusters nc is considered.

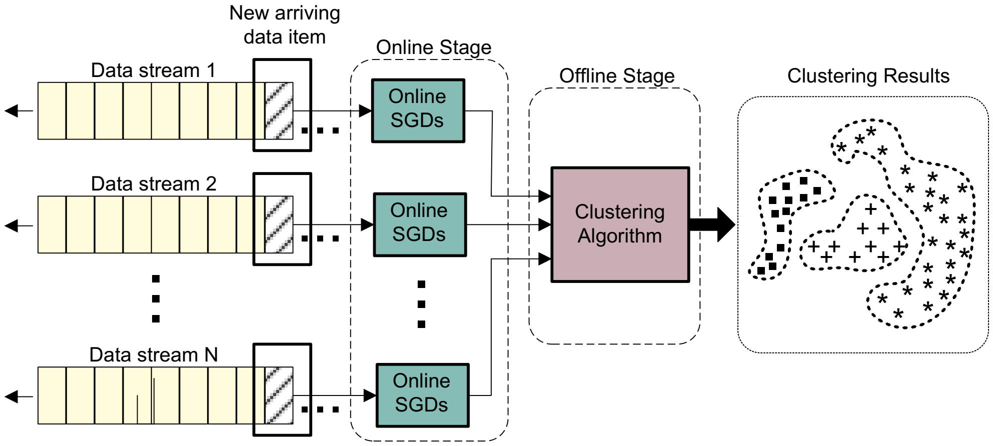
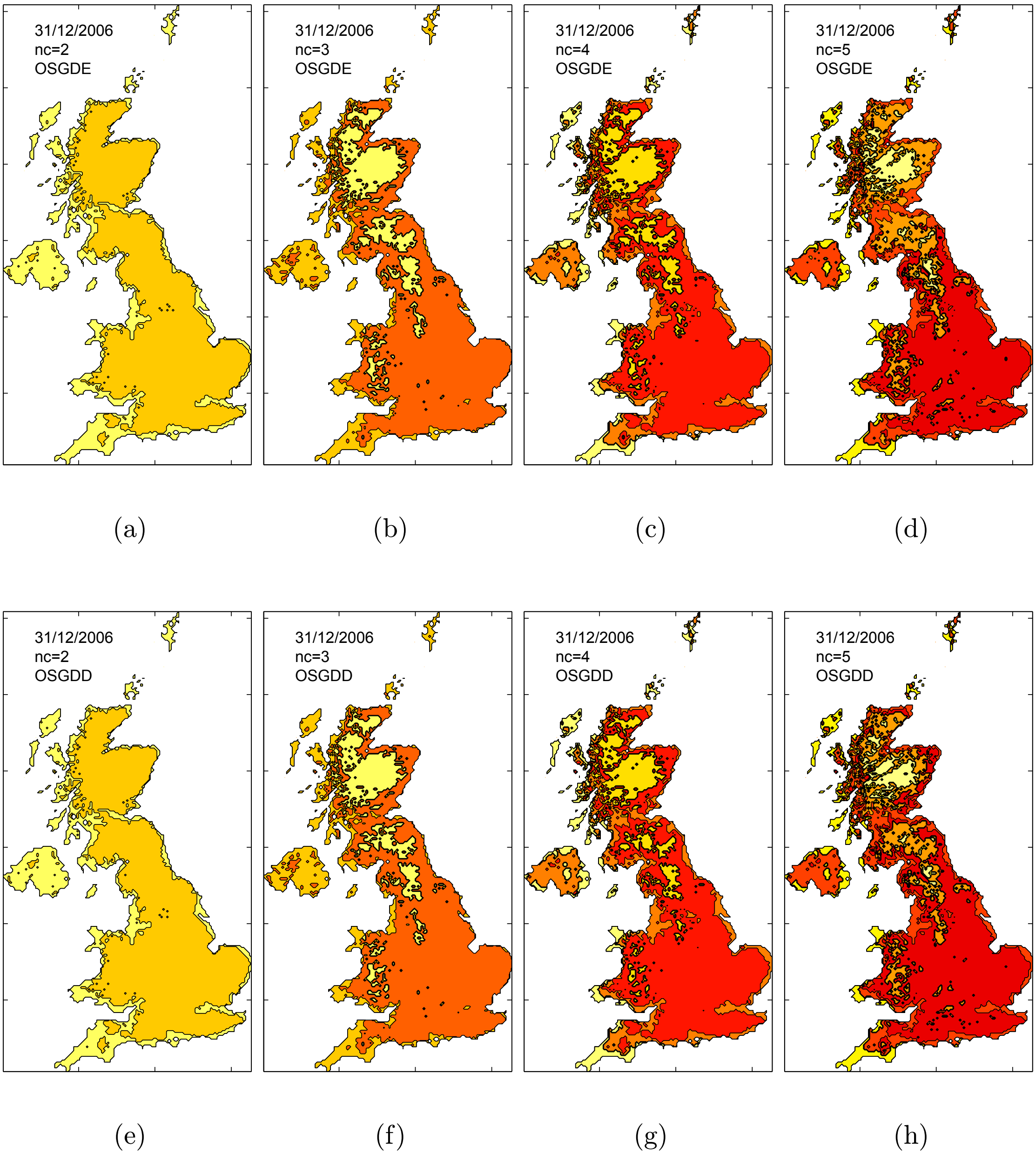


FIGURE 4.28: The proposed OSGD-based data streams clustering framework.

This section presents maps showing the clustering results for both OSGD algorithms obtained when: 1) the parameter nc is varied (Figure 4.29), 2) when the time stamp is different (Figure 4.30), and 3) when the parameters of the OSGDs algorithms are modified (Figure 4.31). Note that since the size of the window is set to 365 for the above described three cases and the data is sampled on a daily based then the maps depicted in Figure 4.29, Figure 4.30 and Figure 4.31, which were obtained for the last day of a given year, are in fact the OSGDs representations that consider all the data for the year in question. Note also that the parameter j_0 is set to 3 for all experiments

The first observation from Figure 4.29 is that the clustering results obtained using OSGD-D and OSGD-E are similar for same number of clusters nc evaluated. Specifically, there is 97.91%, 95.34%, 95.95% and 92.23% correlation between the maps corresponding to $nc = 2$, 3, 4 and 5, respectively. The second observation is related to the fact that when increasing the number of clusters to 3, 4 or 5 (Figure 4.29b, Figure 4.29c and Figure 4.29d, as well as Figure 4.29f, Figure 4.29g, and Figure 4.29h) it can be noted that there is more cluster diversity in the northern regions of the country. On the contrary, locations associated with the southern part of the UK are largely associated with same cluster for the different values of nc investigated. The above results means that the time series representations obtained for the southern locations in the UK are more similar among each other than the ones related to northern locations.



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 4.29: Clustering results for daily temperatures in the UK at a particular date using both OSGDE and OSGDD algorithms with $w = 365$, $M = 1$ and different values for the number of clusters nc.

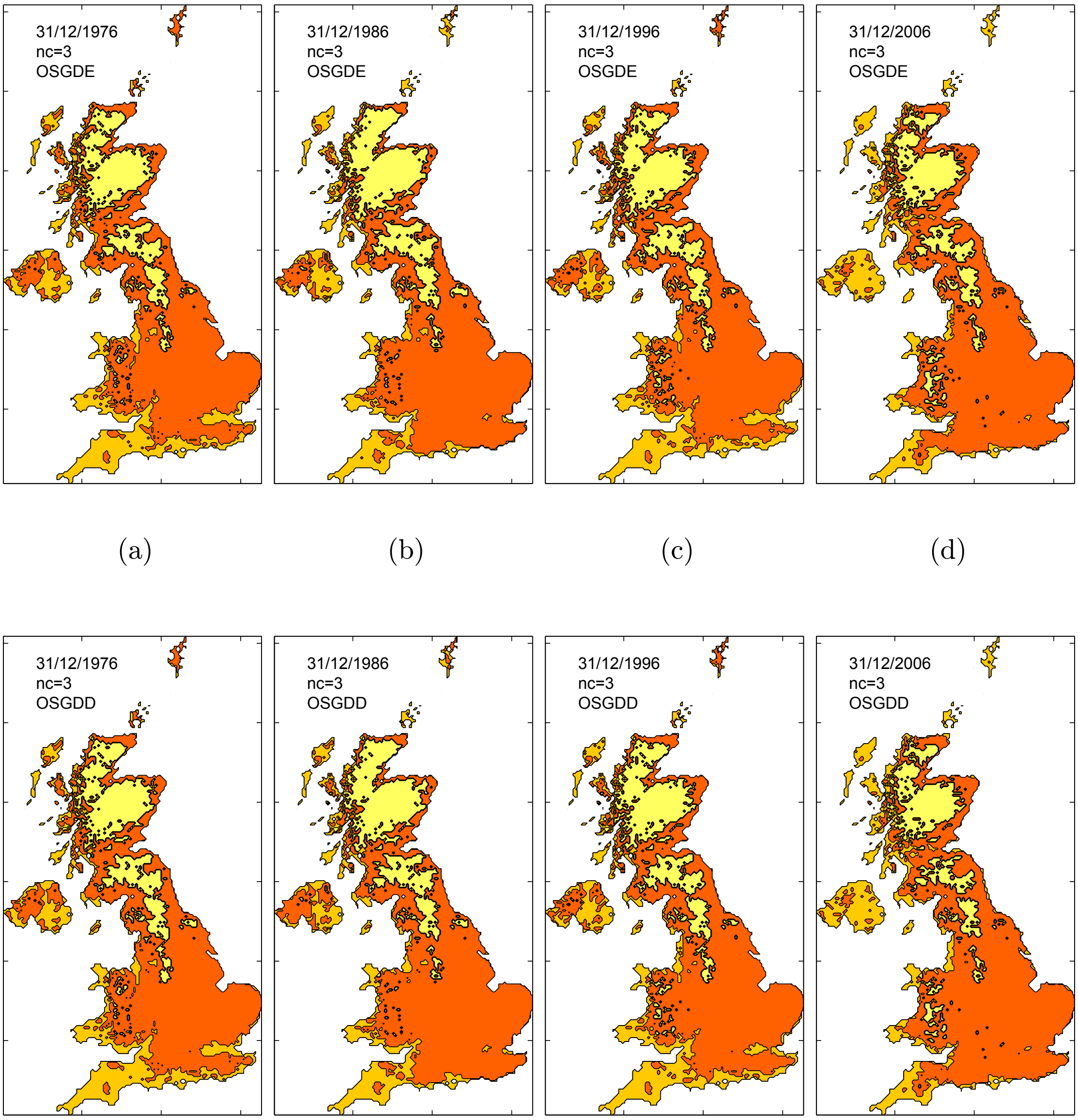
Figure 4.30 shows clustering results for both algorithms evaluated at four specific days keeping nc constant. According to this figure results obtained using OSGD-D and OSGD-E are very similar to each other with a correlation between maps of 94.81%, 94.20%, 96.33%, 95.34% for the 31/12/1976, the 31/12/1986, the 31/12/1996, and the 31/12/2006 respectively. Note also that the resulting clusters are similar among the four specific dates investigated. This result is clearly the expected one for the temperature in a particular region since, independently of the date, there will be always a large degree of spatial correlation between temperatures related to contiguous locations.

Clustering results for different values of the parameter M in OSGD-D and OSGD-E algorithms are depicted in Figure 4.31. Two are the main observations from this figure. The first one is that in general, for both algorithms, results obtained for the two values of M evaluated are largely similar, which means that in this particular application one level of decomposition ($M = 1$) in either the ODWT or MREWMA stages of OSGDs algorithms will bring acceptable clustering results. The second observation is that, for this particular experiment, similar to what happens for the assessments related to Figure 4.29 and Figure 4.30, OSGD-D and OSGD-E algorithms with the same values for parameters M and j_0 produce similar clustering results. In this case, there is 95.10% and 95.86% correlation between OSGD-D and OSDG-E when $M = 1$ and when $M = 2$, respectively.

Finally, it is important to highlight, that the above experiments confirm, at least in an empirical way, that the proposed OSGDs representations when used in clustering applications involving sensor data, will bring coherent and meaningful information.

4.3 Final Remarks

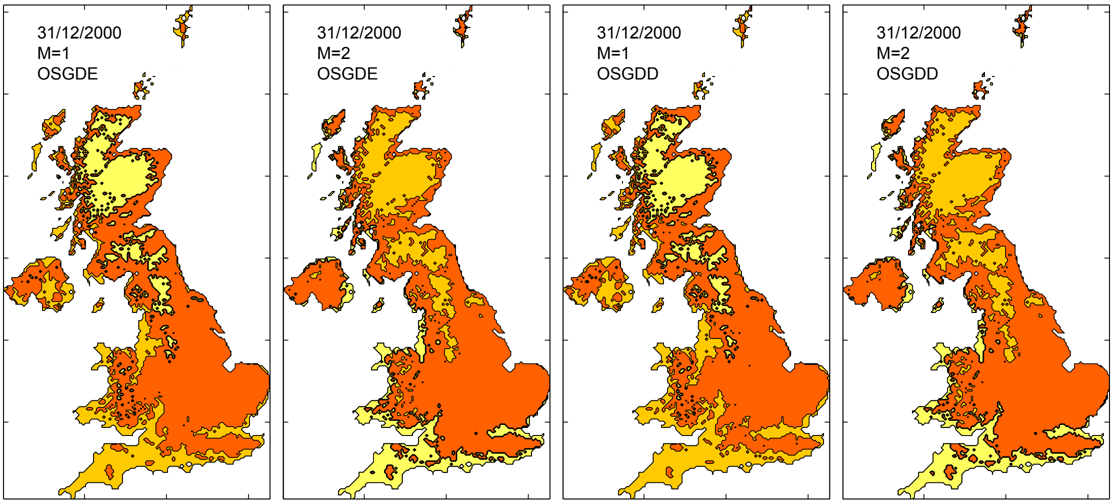
In this chapter, the empirical evaluation of two offline and two online algorithmic instantiations for the proposed SGDs framework was presented. Synthetic data as well ass real world data are considered for the evaluation. The real world data used for the offline algorithms includes bearing vibration data from an inductor motor, ECG data from 16 volunteers, as well as a collection of 42 benchmark time series from diverse disciplines used as benchmark for the time series mining research community. For online algorithms the real world data selected includes air pollution data from Hong Kong and Temperature data from the UK.



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 4.30: Clustering results for daily temperatures in the UK at different dates using both OSGDE and OSGDD algorithms with $w = 365$, $M = 1$ and $nc = 3$.



(a) (b) (c) (d)

FIGURE 4.31: Clustering results for daily temperatures in the UK at a particular date using OSGD-E and OSGD-D algorithms with different values of M and with $w = 365$, $nc = 3$ for both methods.

The results reported for the two offline experiments show that the new time series data representation framework here proposed outperforms the 21 benchmark time series representation included in the evaluation. Regarding the third offline experiment, the algorithms based on the proposed framework reports the best outcome in 33.8% of the data sets evaluated considering all distance measures studied. In contrast, the best benchmark algorithm reports the best outcome in only 11.9% of the cases. Moreover, by considering each distance measured separately, the proposed SGDW3 algorithm reports the larger number of best outcomes in 4 out of 5 distances.

The online algorithms have been assessed in the context of two data mining tasks namely, change detection and multiple data streams clustering. The results show the potential of the proposed online SGDs framework as a promising alternative to traditional online data stream processing algorithms. In a great variety of systems and processes different types of events, related to particular states of operation or behaviour, need to be discovered as early as possible. In this context, online algorithms such as the proposed OSGDs representations, which help to provide online information about emerging events, can be of immense value for the decision making process.

The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas, like the colors or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in this world for ugly mathematics.

Godfrey H. Hardy,
A mathematician's apology (1940)

Chapter 5

Data Stream Evolution Diagnosis using RWDE

A plethora of algorithms have been designed for mining streaming data; however, only few approaches have been reported for the online characterisation of its evolution. In this chapter, a novel framework for diagnosing the evolution of multidimensional streaming data which incorporates a recursive wavelet-based density estimation approach is proposed. In the proposed framework changes in streaming data are characterised by the use of *local* and *global evolution coefficients*. In addition to this, for the analysis of changes in the correlation structure of data, a recursive implementation of Pearson's correlation coefficient using exponential discounting is proposed. Visualisation tools, such as temporal and spatial velocity profiles [9], are here extended to fit into the proposed framework.

This chapter is organised as follows. Section 5.1 presents a brief introduction to the problem of evolution detection in the context of streaming data. In Section 5.2, theoretical background regarding RWDE is briefly reviewed. In Section 5.3 the proposed data stream evolution diagnosis framework is presented. Section 5.4 includes the empirical evaluation of the framework. Final remarks are presented in Section 5.5.

5.1 Introduction

In streaming data analysis, data evolution refers to the process in which important changes occur over time in the trends of a given data stream due to changes in the underlying phenomena [9]. Algorithms for the diagnosis of evolution in data streams assume that, by analysing changes in data, valuable insights can be obtained to characterise the phenomena under study.

Although several algorithms can be found in the literature dealing with the problem of data stream mining, regarding the diagnosis of data evolution, the most relevant work to the method presented in this chapter is the concept of *velocity density* proposed by Aggarwal in [9] which measures the rate of change of data concentration at a given spatial location over a user-defined time horizon.

In this chapter, a novel evolution diagnosis framework for data streams is proposed. This framework is based on extending the concept of *velocity density estimation*, originally introduced within the context of Kernel Density Estimation (KDE), to RWDE. The proposed framework has some important advantages with respect to the approach reported in [9]: 1) it requires a significant lower amount of memory; 2) the proposed algorithms are computationally less complex; and 3) for higher dimensional data, it allows the diagnosis of data evolution separately for each dimension with only one pass of the data.

Although it has been previously suggested that data evolution can be estimated by means of velocity density estimation, the proposed method is fundamentally different from [9] in the following aspects. The first difference is the method selected for the estimation of probability densities. Note that the proposed framework uses a RWDE optimised for online applications instead of an off-line KDE. The second difference is that in [9], the velocity density is formulated in terms of sliding windows whereas in the proposed framework it considers exponential discounting. Moreover, in [9] to estimate velocity density profiles and in general to quantify data evolution two types of densities are considered namely, a *forward density estimate* and a *reverse density estimate.* The main idea in [9] is to compare a density that assigns more importance to old data with one that puts more emphasis on new data. Note that in the proposed method the concepts of reverse and forward density do not apply and instead of that a pair of density estimates related to different exponential discounting/updating strategies for the coefficients of each estimator is used. The last important difference is that in the proposed framework velocity densities are formulated in a separate way for each dimension. This capability not only allows the localised diagnosis of data evolution in each dimension, but also it provides the basis for the detection of particular dimensions or combinations of dimensions in the data which are relevant for a given data change.

In the proposed framework, the characterisation of changes in streaming data relies on concepts such as evolution coefficients and correlation between evolution coefficients. An *evolution coefficient* is an indicator of the level of significance of data changes at each particular time stamp. Here a high level of evolution is associated with significant changes of data concentrations at various spatial locations over a particular time horizon. The correlation coefficient between the evolution coefficients related to different dimensions captures changes in the correlation structure of the data. Specifically, a recursive implementation of the Pearson's correlation coefficient based on exponential discounting is proposed. The two visual profiles proposed in [9] relying on the concept of velocity density namely, *temporal velocity profiles* and *spatial velocity profiles*, are also extended here to the context of proposed RWDE-based VDE framework.

5.2 Theoretical Background

5.2.1 Recursive Wavelet Density Estimator (RWDE) Overview

5.2.1.1 Batch WDE

The main idee of WDEs, which were introduced in Chapter 2, is that an unknown square integrable density function $f(x)$, with $X_1, X_2, ..., X_n$ denoting the realisations of a random variable X, can be expressed as the convergent series of orthogonal basis functions in $\mathbf{L}^2(\mathbb{R})$. Depending on the type of basis functions employed (only scaling functions or scaling and wavelet functions), in WDEs two main alternatives for the density can be formulated. Since data streams applications are constrained by computational restrictions, in this chapter the simplest WDE is considered. This linear estimator is defined by:

$$\hat{f}(x) = \sum_k \hat{c}_{j_0,k}\phi_{j_0,k}(x) \tag{5.1}$$

where $\phi_{j_0,k}(x) = 2^{-j_0/2}\phi(2^{-j_0}x - k)$ is the scaling function associated with the *base resolution* 2^{j_0} with the index $j_0 \in \mathbb{Z}$ and $k \in \mathcal{K} \subset \mathbb{Z}$. In order to simplify notation, this work considers scaling function filters whose support at resolution $2^{-j_0} = 0$ is $2n_\phi - 1$ with n_ϕ denoting the order of the filter. The well known Daubechies and Symmlets wavelet families include scaling functions with these characteristics. Assuming that each X_i is normalised to take only values within the interval $[0, 1]$, and considering $[2^{j_0}k, 2^{j_0}(k + 2n_\phi - 1)]$ to be the support of $\phi_{j_0,k}$, then $\mathcal{K} = \{-(2n_\phi - 2), \ldots, 0, \ldots, (2^{j_0} - 1)\}$ (the reader is referred to [96] for more details). In Equation (5.1), coefficients $\hat{c}_{j_0,k}$ are estimated according to

$$\hat{c}_{j_0,k} = \frac{1}{n}\sum_{i=1}^{n}\phi_{j_0,k}(X_i) \tag{5.2}$$

It has been shown in [74] that the estimator $\hat{f}(x)$ is suboptimal. However, since in the proposed framework the interest is placed on the fast evaluation of the relative difference between densities rather than in their precise estimation, then the prompt detection of relative changes in the density will be very useful. Note that the density estimation process in the linear WDE of Equation (5.1) is in fact a projection into the space $\mathbf{V}_{j_0}$ and as a result the density obtained is an approximate version of the true density at resolution 2^{-j_0}. For further details about this type of estimator the reader is referred to Chapter 2 and to [74].

5.2.1.2 Extension to higher dimensions

The estimator described above can be easily extended to higher dimensions by considering multidimensional multiresolution analysis and their corresponding multidimensional wavelets. In this case, by considering $\mathbf{X}_i = (X_i^1, X_i^2, \ldots, X_i^m) \in \mathbb{R}^m$ to be the realisations of a multidimensional random variable $\mathbf{X}$ and using $\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m$ then the m-dimensional density can be expressed as:

$$\hat{f}(\mathbf{x}) = \sum_{\mathbf{k}} \hat{c}_{j_0,\mathbf{k}}\Phi_{j_0,\mathbf{k}}(\mathbf{x}) \tag{5.3}$$

where, $\mathbf{k} = (k_1, k_2, \ldots, k_m) \in \mathbb{Z}^m$ and $j_0 \in \mathbb{Z}$. The m-dimensional approximation of the scaling coefficients can be extended from Equation (5.2) and can be expressed as:

$$\hat{c}_{j0,\mathbf{k}} = \frac{1}{n}\sum_{i=1}^{n} \Phi_{j0,\mathbf{k}}(\mathbf{X}_i) \tag{5.4}$$

where the m-dimensional basis function is defined by $\Phi_{j_0,\mathbf{k}}(\mathbf{x}) = \bigotimes_{d=1}^{m} \phi_{j_0,k}(x_d)$ with the symbol $\bigotimes$ denoting tensor product. Note that Equation (5.3) and Equation (5.4) are based on the concept of *separable multiresolution approximations* [185], in which m-dimensional scaling functions are defined as the tensor product of m one-dimensional scaling functions.

5.2.1.3 Recursive WDE

For the online implementation of the WDE the recursive estimator proposed by [157], referred to as RWDE, is used. This estimator originally comprises two stages. The initial stage is an off-line stage in which an initial estimate of the density is obtained using the batch WDE of Equation (5.1) and Equation (5.2). In the second stage, which is an online one, the estimator's coefficients $\hat{c}_{j0,k}(t)$ are recursively updated as new data items arrive according to the equation:

$$\hat{c}_{j0,k}(t) = (1-\theta)\hat{c}_{j0,k}(t-1) + \theta\phi_{j0,k}(X_t) \tag{5.5}$$

where X_t, the newest data item available, denotes the realisation of a random variable X at time t. Since coefficients $\hat{c}_{j0,k}$ are recursively estimated each time a new data item arrives, the temporal index t is included. Note that Equation (5.5) defines an exponential discounting strategy for the estimator coefficients where the parameter θ controls the emphasis assigned to new data respect to older one[1].

It is important to point out a specific issue of the updating strategy described by Equation (5.5). When the newest data item available X_t falls outside the support of the scaling function $\phi_{j0,k}$, the product $(1-\theta)\phi_{j0,k}(X_t)$ is zero and then the updating of coefficient $\hat{c}_{j0,k}(t)$ does not yield the expected results. Note that in such circumstances, $\hat{c}_{j0,k}(t)$ should be equal to its past value. In order to fix this problem it is proposed to use the selective coefficients evaluation method reported in [96]. In this way, prior to the updating of $\hat{c}_{j0,k}(t)$ it is checked if X_t falls within the interval $[2^{j_0}k, 2^{j_0}(k+2n_\phi-1)]$, which is the support of $\phi_{j0,k}$. If X_t is inside this interval then Equation (5.5) is used; otherwise, the coefficient is

[1] In order to find a simple direct relation between θ and the window size parameter used, for the weight assigned to the previous value of the coefficients $c_{j_0,k}$, in this chapter the term $(1-\theta)$ is used instead of θ.

left unchanged. For further details about this procedure and the implementation of RWDE the reader is referred to [96].

5.2.2 The Velocity Density Estimation Framework

The velocity density concept, first introduced in [9] in the context of KDE and sliding windows, is based on the intuitive idea that high levels of evolution in a given data stream are associated with changes in relative data concentrations at some given spatial locations. According to this, such changes can be captured by estimating the variations in the probability density of the data in a given spatial location over time. Specifically, the concept of velocity density relies on the difference between two probability densities, each of them associated with a particular temporal weighting strategy for the data covered by the sliding window $(t - h_t, t)$, where h_t defines the length of the temporal window. For this purpose in [9] a *forward time slice density estimate* and a *reverse time slice density estimate* are employed. While the former is related to the probability density for all data items covered by the sliding window giving a higher importance to new data, the latter relies on computing the probability density emphasizing the importance of old data. Hence, the larger the difference between these two densities the higher the amount of change in the data stream at that given time t. Additionally, since different values for h_t are related to different time horizons for the analysis, with large values of h_t, long-trends can be investigated whereas by using a small h_t, short-term trends can be studied.

Note that, in its original formulation, the computation of velocity density requires the storage in memory of all data items covered by the sliding window. Also note that in [9] the probability densities are estimated using KDE, which require the same number of Gaussian functions than the number of data items used for the estimation. It is evident then that if h_t is chosen to be large, not only the amount of memory required will increase but also the computational burden of the algorithm will be higher. An additional burden of the approach reported in [9], is that for the case of high dimensional data, the evaluation of the amount of evolution at each particular dimension involves scanning the data more than once. The above three disadvantages prevent the potential deployment of the framework in real world applications.

In order to overcome the above issues and in order to provide an intuitive user-friendly data stream evolution algorithm, a novel method for the estimation of velocity density is proposed in this chapter. The proposed method uses a different density estimator and relies on a rather different data discounting strategy.

5.3 Proposed RWDE-based Velocity Density Estimation Framework

There are various reasons why RWDE is chosen as the estimator in the proposed velocity density estimation framework. Firstly, this algorithm does not require the storage of past data items in memory. In addition to this, it employs a fixed number of basis functions, independent of the number of data items used for the estimation. Furthermore, the use of this particular type of estimator not only allows the evaluation of data evolution at each dimension separately, but it also renders a radically different and computationally less expensive framework for the estimation of local and global evolution coefficients.

It is important to point out a fundamental conceptual difference of the proposed RWDE-based framework respect to the approach reported in [9]. In [9], the evolution diagnosis framework is based on velocity densities which are constructed by comparing two probability density functions estimated using one sliding window but applying different temporal weighting strategies for the data covered by the window. Hence, the key idea in [9] is the use of a spatiotemporal kernel function that is a time-factored version of the spatial kernel traditionally used in the KDE context. In contrast, the proposed density estimation framework relies on the use of two recursive exponential windows which, for the updating of the density estimates at a given time stamp, only require the last data item available. Following this approach the difference between two probability densities can be captured by applying different exponential discounting strategies to old data and new data.

The main advantage of the proposed method is that it requires constant time for the computation of the densities, independent of any window size. Note here that using the framework reported in [9] the quantification of long-term changes in data is performed by increasing the size of the sliding window. Consequently, the amount of memory required and the computational complexity of the algorithm increases. In contrast, in the proposed approach, long-term trends in data can be analysed without increasing neither memory storage nor computational burden.

The two approaches for velocity density estimation are depicted in Figure 5.1. Note that the two main differences are: 1) the data used for the estimation of the density; and 2) the type of window used for the estimation. While the method shown in Figure 5.1a requires all the data that falls inside the sliding window, the technique shown in Figure 5.1b works only with one data item. Regarding the window employed, we can see that the weighting strategy for approach reported in [9], as shown in Figure 5.1a, is in fact a linear one; whereas for the proposed approach (Figure 5.1b) is of an exponential nature.

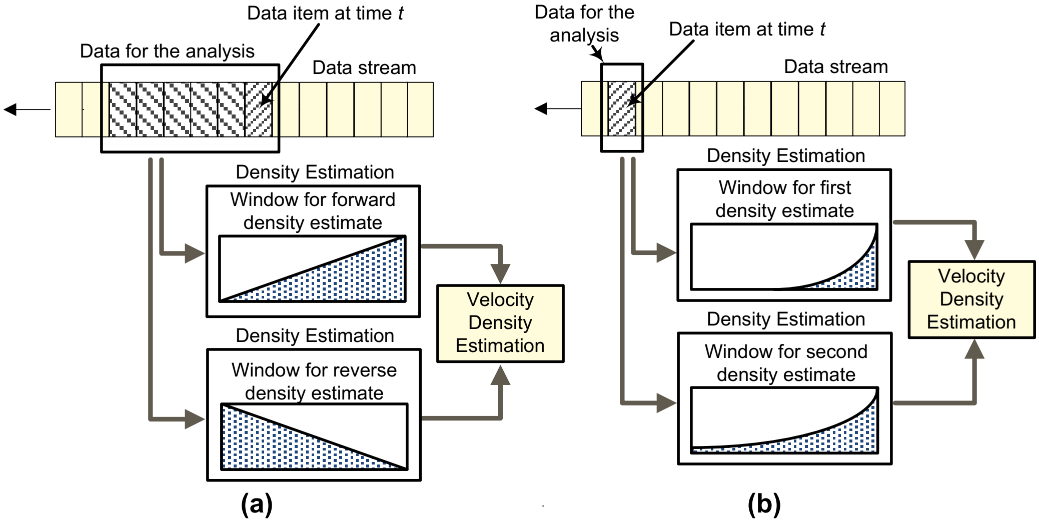

Data item at time *t*
Data for the analysis
Data stream
Density Estimation
Window for forward density estimate
Velocity Density Estimation
Density Estimation
Window for reverse density estimate
(a)
Data for the analysis
Data item at time *t*
Data stream
Density Estimation
Window for first density estimate
Velocity Density Estimation
Density Estimation
Window for second density estimate
(b)

FIGURE 5.1: Windows used in the evolution diagnosis frameworks. (a) Framework reported in [9]; (b) Proposed RWDE-based VDE framework.

5.3.1 Proposed Velocity Density Framework

The algorithms proposed in this section consider a multivariate data stream similar to the one defined in Section 5.2.1.2 whose multidimensional data items at time t, $\mathbf{X}_t = (X_t^1, X_t^2, \ldots, X_t^m) \in \mathbb{R}^m$, are assumed to be the realisations of a multidimensional random variable $\mathbf{X}$.

5.3.1.1 Density Estimation

In the proposed framework, which follows the RWDE concepts reviewed in Section 5.2.1, velocity densities are calculated using a pair of m-dimensional probability density estimates of the form:

$$\hat{f}_\theta(\mathbf{x}, t) = c_{norm} \bigotimes_{d=1}^{m} \hat{f}_\theta^d(x_d, t) \tag{5.6}$$

where the symbol $\bigotimes$ denotes the tensor or outer product of two vectors, θ defines a given discounting strategy, $d \in \mathcal{D} = \{1, 2, \ldots, m\}$ is index related to the dimension of the data and c_{norm} is a normalisation constant to make $\int_x \hat{f}_\theta(\mathbf{x}, t) = 1$. Note that here $\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m$. In Equation (5.6), each one-dimensional density is in turn expressed as:

$$\hat{f}_\theta^d(x_d, t) = \sum_k \hat{c}_{j_0,k}^d(t)\phi_{j_0,k}(x_d) \tag{5.7}$$

with the coefficients of the estimator recursively updated each time a new data item is available according to:

$$\hat{c}_{j_0,k}^d(t) = (1-\theta)\hat{c}_{j_0,k}^d(t-1) + \theta\phi_{j_0,k}(X_t^d) \tag{5.8}$$

where X_t^d refers to the last data item available at time t and dimension d. Here, $k \in \mathcal{K} = \{-(2n_\phi - 2), \ldots, 0, \ldots, (2^{j_0} - 1)\}$.

Note that in Equation (5.6), Equation (5.7) and Equation (5.8) the temporal variable t is introduced to both, density estimates and coefficients, to indicate their time dependency. According to the above equations, the procedure to estimate a given multidimensional density $\hat{f}_\theta(\mathbf{x}, t)$, at each time stamp t is performed in two steps. The first step estimates the m one-dimensional densities using Equation (5.7). The second step combines the one-dimensional densities using Equation (5.6).

As it was described in Section 5.2.1, the RWDE algorithm originally comprises two stages namely, an off-line and an online stage. Since in data stream applications the estimation of an initial density estimate is not feasible, in the RWDE implementation proposed the estimator's coefficients are initialised to zero, that is, $\hat{c}_{j_0,k}^d(0) = 0 \; \forall k \in \mathcal{K}$. Note also that for the evaluation of Equation (5.6) and Equation (5.7) the same set of P points $\beta = \{b_P\}_{p \in 1,2,\ldots,P}$, with $b_P \in [0, 1]$, is needed for all dimensions. Since RWDE involves two general stages, that is, updating of estimator's coefficients and evaluating the density in a set of P points β, the pseudo code has been split into Algorithm 5.1 and Algorithm 5.2, respectively. Note that these algorithms are related to the one-dimensional case.

Algorithm 5.1: RWDE_updating(j_0,X_t,θ,$\{\hat{c}_{j_0,k}(t-1)\}_{k\in\mathcal{K}}$)

```
1
Input: $j_0$: Resolution of scaling functions; $X_t$: newest data item; $\theta$: Discounting parameter;
       $\{\hat{c}_{j_0,k}(t-1)\}_{k\in\mathcal{K}}$: Scaling function coefficients at time $t-1$;.
Output: $\{\hat{c}_{j_0,k}(t)\}_{k\in\mathcal{K}}$: Updated scaling function coefficients at time $t$.
for $k \leftarrow -(2n_\phi - 1)$ to $2^{j_0}$ do
    if $X_t \geq 2^{j_0}k$ and $X_t \leq 2^{j_0}(k + 2n_\phi - 1)$ then
        $\hat{c}_{j_0,k}(t) = (1-\theta)\hat{c}_{j_0,k}(t-1) + \theta\phi_{j_0,k}(X_t)$;
    else
        $\hat{c}_{j_0,k}(t) = \hat{c}_{j_0,k}(t-1)$;
    comment: If the new arriving data item $X_t$ falls within the support of the scaling function
    then update the scaling function coefficient otherwise make it equal to its past value.
```

Algorithm 5.2: RWDE_evaluation(j_0,$\{c_{j_0,k}(t)\}_{k\in\mathcal{K}}$,$\beta$)

```
1
Input: $j_0$: Resolution of scaling functions; $\{c_{j_0,k}(t)\}_{k\in\mathcal{K}}$: Updated scaling function coefficients at
       time $t$; $\beta = \{b_p\}_{p\in\{1,2,\ldots,P\}}$: Set of points for the evaluation of the density.
Output: $\{\hat{f}^d(X_\beta, t)\}_{X_\beta\in\beta}$: Updated probability density function at time $t$.
for $X_\beta \leftarrow b_1$ to $b_P$ do
    for $k \leftarrow -(2n_\phi - 1)$ to $2^{j_0}$ do
        $\hat{f}^d(X_\beta, t) = \hat{c}_{j_0,k}(t)\phi_{j_0,k}(X_\beta)$;
        comment: Evaluate the probability density estimate at a grid of points $X_\beta \in \beta$.
```

Since the velocity density framework is based on the quantification of the difference between two densities, in the proposed approach a pair of RWDE with different values for θ is required. The procedure to obtain the appropriate pair of values for θ is described in the the following section.

5.3.1.2 Choice of parameter θ

The procedure for the selection of the parameter θ involves making the weighting value of a particular past coefficient equal to a user defined parameter c_{min}, related to the desired weighting value for coefficient $\hat{c}_{j_0,k}(t-w)$, where w is an hypothetical window length, with $0 < c_{min} < 1$, $w > 1$ and $w \in \mathbb{N}$. In order to accomplish this, first consider that Equation (5.5) can be expanded in the following way:

$$\begin{aligned}\hat{c}_{j_0,k}(t) = \quad & (1\text{-}\theta)^t\hat{c}_{j_0,k}(0) \\ & + (1\text{-}\theta)^{t-1}\theta\hat{c}_{j_0,k}(1) + (1-\theta)^{t-2}\theta\hat{c}_{j_0,k}(2) \\ & + \ldots (1\text{-}\theta)\theta\hat{c}_{j_0,k}(t-1) + \theta\phi_{j_0,k}(x(t))\end{aligned} \tag{5.9}$$

According to Equation (5.9) the weight assigned to coefficient $\hat{c}_{j_0,k}(t-w)$, that is the last coefficient covered by an hypothetical window of length w, is $(1-\theta)^{t-(t-w)}\theta$. Taking this into account, θ can be estimated from the following equation:

$$\begin{aligned} (1\text{-}\theta)^{t-(t-w)}\theta &= \mathrm{c}_{min}(1-\theta)\theta \\ (1\text{-}\theta)^{w-1} &= \mathrm{c}_{min} \\ \theta &= 1\text{-}(\mathrm{c}_{min})^{1/(w-1)} \end{aligned} \tag{5.10}$$

For normalisation purposes, in Equation (5.10) it is considered that weighting value for the coefficient $\hat{c}_{j_0,k}(t-w)$ should be equal to c_{min} times the weighting value of coefficient $\hat{c}_{j_0,k}(t-1)$ which is $(1-\theta)\theta$. The experiments of Section 5.4 consider $c_{min} = e^{-1} = 0.3679$ which means that the weighting value corresponding to the coefficient obtained at time $t-w$ (that is, the oldest element in the hypothetical window) has a contribution of 36.79%. It is easy to see that by choosing this particular value for c_{min}, θ becomes approximately equal to $1/w$.

The characteristics of different discounting strategies are shown in Figure 5.2, where the weighting values assigned at time stamp 900 to the coefficients related to previous time stamps using three different values of θ is plotted. Note that in Figure 5.2 the curves are normalised using $(1-\theta)\theta$ equal to 1. Regarding the notation, the value of θ associated with a particular exponential window w is denoted as θ_w.

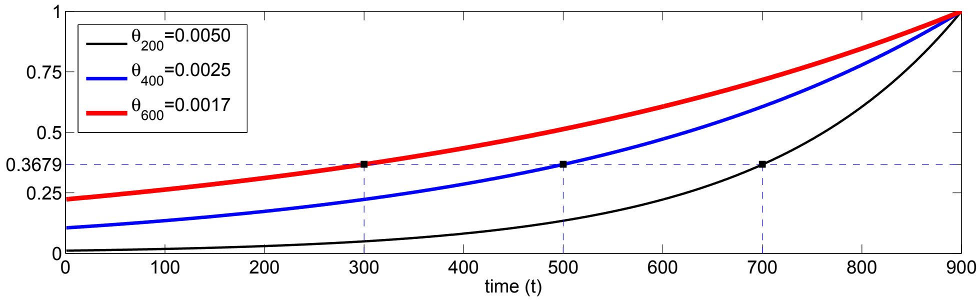


FIGURE 5.2: Weighting strategies for different values of θ.

Since in the proposed velocity density framework two densities are required, each of them related to different discounting strategies, Equation (5.10) is evaluated using two different window lengths, w_1 and w_2. Then the notation θ_{w1} and θ_{w2} is used to refer the discounting parameters associated with w_1 and w_2, respectively. Note that $\mathrm{w}_1, w_2 \in \mathbb{N}$, are user defined quantities whose values depend on the desired time horizon for the analysis. Therefore, if $w_2 > w_1$, then w_1 is related to a hypothetical window that includes less number of data items for the density estimation than the ones included by the window associated with w_2. Here, $w_2 - w_1$ implicitly expresses a given time horizon for the analysis. In this work, in order to achieve results similar to the ones obtained using the velocity density estimation framework of [9], it is proposed that w_1 should be the quarter of w_2, that is, $w_1 = w_2/4$. This selection would guarantee that the associated densities would significantly differ from each other when some changes in the last w_2 items are present. Note that w_2 is the largest window considered.

5.3.1.3 Velocity Density Estimation

In this section two different strategies for the estimation of the velocity density are suggested. The first method is mainly useful for visualisation purposes. The second approach is more appropriate for the fast computation of local an global evolution coefficients.

Velocity Density Estimation from Densities (VDD)

The first strategy for velocity density estimation is based on evaluating the difference between two density estimates of the data using the densities themselves. This method is referred to as Velocity Density estimation from Densities (VDD). In this case, by using the one-dimensional densities of Equation (5.7), the one-dimensional velocity density $V^d_{(\theta_{w_1},\theta_{w_2})}(X^d,t)$ at location X^d and time t for dimension d is defined by:

$$V^d_{(\theta_{w_1},\theta_{w_2})}(X^d,t) = \frac{\hat{f}^d_{\theta_{w1}}(x,t) - \hat{f}^d_{\theta_{w2}}(x,t)}{w_2 - w_1} \tag{5.11}$$

where θ_{w1} and θ_{w2} are two different exponential discounting strategies, with w_1 and w_2, referring to the window lengths used for the estimation.

The velocity density for the entire set of dimensions $V_{(\theta_{w_1},\theta_{w_2})}(X,t)$ at spatial location X and time t can be expressed as:

$$V_{(\theta_{w_1},\theta_{w_2})}(X,t) = \frac{\hat{f}_{\theta_{w1}}(\mathbf{x},t) - \hat{f}_{\theta_{w2}}(\mathbf{x},t)}{w_2 - w_1} \tag{5.12}$$

Note in Equation (5.12) $\hat{f}_{\theta_{w1}}(\mathbf{x},t)$ and $\hat{f}_{\theta_{w2}}(\mathbf{x},t)$ are multidimensional densities. Since the total volume under each of the probability densities involved in Equation (5.11) and

Equation (5.12) is equal to the unit, then the maximum volume contained in the VDD is 2. Note also that here the term spatial location refers to a specific point inside the support of either a given multidimensional density or a given multidimensional velocity density. It is also important to highlight that one of the advantages of the above velocity density formulation, is that it allows the use of different discounting strategies for each dimension.

Velocity Density Estimation from Coefficients (VDC)

Since the main objective of the velocity density framework is the quantification of the rate of change in data over a particular time horizon and by considering that in the context of RWDE the way a given density varies over time is directly related to the way its corresponding parameters change; then an alternative velocity related to the change in the density estimator's coefficients can be used to realise an alternative formulation. This strategy is called Velocity Density estimation from Coefficients (VDC).

This velocity density estimation strategy has the advantage of a reduced computational burden since it is directly evaluated from the estimator's coefficients, avoiding the evaluation, at every time stamp t, of the density (using either Equation (5.6) or Equation (5.7)) in a set of points β. This alternative velocity can be expressed, for the one-dimensional case, as:

$$\hat{V}^d_{(\theta_{w_1},\theta_{w_2})}(k,t) = \frac{\hat{\mathbf{c}}^d_{\theta_{w1}}(k,t) - \hat{\mathbf{c}}^d_{\theta_{w2}}(k,t)}{w_2 - w_1} \tag{5.13}$$

where $\hat{\mathbf{c}}^d_{\theta_{wx}}(k,t) = [\hat{c}^d_{j0,-2n_\phi-1}(t),\ldots,c^d_{j0,2^{j0}}(t)]$ is a vector containing the $(2^{j_0} + 2n_\phi - 2)$ estimator's coefficients using the discounting strategy defined by w_x. Note here that the velocity defined by Equation (5.13) is a discrete function since k is a discrete set of translation parameters for the scaling function $\phi_{j0,k}$.

The VDC for the entire set of dimensions is defined as:

$$\hat{V}_{(\theta_{w_1},\theta_{w_2})}(\mathbf{k},t) = \frac{\hat{\mathbf{c}}_{\theta_{w1}}(\mathbf{k},t) - \hat{\mathbf{c}}_{\theta_{w2}}(\mathbf{k},t)}{w_2 - w_1} \tag{5.14}$$

where $\hat{\mathbf{c}}_{\theta_{wx}}(k,t) = \bigotimes_{d=1}^{m} \hat{\mathbf{c}}^d_{\theta_{wx}}(k,t)$ is the tensor product of the d one-dimensional vector of coefficients. Note that in order to compute velocity densities directly from the estimator's coefficients the same base resolution 2^{-j_0} is used for each of the vectors of coefficients involved.

5.3.2 Visualisation of Changes in 2-dimensional Data

In this section, the concepts of *Temporal Velocity Profiles* (TVPs) and *Spatial Velocity Profiles* (SVPs) introduced in [9], which are tools for visualising changes in two dimensional streaming data, are extended to the RWDE context. Note that in the proposed framework, the construction of TVPs and SVPs is based on VDDs since in VDCs the set of points β is fixed (equal to the number of estimator's coefficients employed) and does not allow the visualisation of a particular spatial location of the data with a higher level of detail.

5.3.2.1 Temporal Velocity Profiles (TVPs)

The temporal velocity profile is defined as the global overview of the velocity density at different spatial points for a given specific time [9]. Hence, this profile is a surface plot of the velocity density using a user defined grid of spatial locations among the two predefined dimensions. Note that this visualisation tool is build up on two dimensional velocity densities.

For the construction of TVPs a discretised version of the velocity density is required. This discrete VDD is obtained by evaluating Equation (5.11) in a set of points $\beta = \{b_P\}_{p \in 1,2,\ldots,P}$. Note that this set contains the P points in which each of the two densities used in the construction of the velocity density are evaluated. By increasing the number of points in β a more detailed TVP is obtained.

5.3.2.2 Spatial Velocity Profiles (SVPs)

This visualisation profile is defined as the spatial overview of the reorganisations that are taken place in the density of the data at specific spatial points [9]. It is useful for getting more insights about how data is shifting from some particular location to another. Note that in this context, a data shift means that the probability of some particular values in the data increases while the probability of other values decreases. data Specifically, SVPs allow the user to easily observe the associated directions of all those changes that are taking place in the data. Intuitively, when data is shifting from one spatial location to another the source of the shift shows a reduction in the probability density whereas the destination of the shift shows an increment. Therefore, there is an increasing gradient from the source to the destination of the shift. The basic idea behind this profile is, firstly, the estimation of the gradient at given grid of spatial locations and then the plotting of the corresponding velocity vectors as arrows at these locations.

For the construction of SVPs a set of points β along each dimension in respect to which the gradient of the velocity density will be evaluated is used. Then for each spatial location X in this grid the velocity gradient along the i-th dimension is defined by

$$\Delta v_i(X,t) = \lim_{\epsilon \to 0} \frac{V_{(\theta_{w_1},\theta_{w_2})}(X + \overline{\epsilon_i}, t) - V_{(\theta_{w_1},\theta_{w_2})}(X,t)}{\epsilon} \tag{5.15}$$

where $\overline{\epsilon_i} = \epsilon.\overline{e_i}$ is an ϵ-perturbation along the i-th dimension and with $\overline{e_i}$ denoting a unit vector. The corresponding TVP and SVP for an example two-dimensional data stream are shown in Figure 5.3. It can be observed from this figure that the two profiles clearly show that there is a change in the statistical properties (mean and variance) of the second dimension of the stream. Both profiles are useful to identify the data locations relevant to this change. In addition to this, SVP also indicate that the main concentration of data moves from 0.5 to 0.7 in the second dimension.

5.3.3 Characterising Data Evolution

In order to characterise if specific trends are occurring at specific data locations the estimation of *local* and *global data evolution coefficients* is proposed. Regarding the characterisation of changes in the correlation structure of the data this is obtained by estimating the *correlation between evolution coefficients.*

5.3.3.1 Evolution Coefficient (EC)

In order to quantitatively address the level of significance of changes in streaming data the evaluation of both *local* and *global data evolution coefficients* using VDC is proposed. Local evolution coefficients (LECs) are related to single dimensions of the data stream whereas the global coefficients (GEC) consider the entire sets of dimensions.

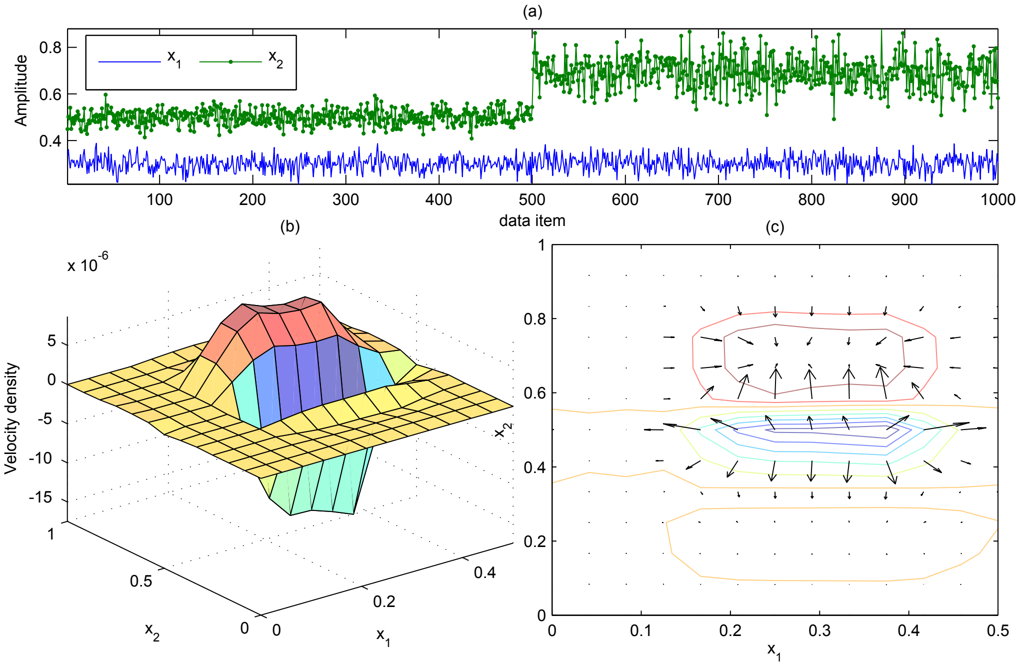


FIGURE 5.3: A two dimensional data stream (a) and its corresponding TVP (b) and SVP (c) representation.

The proposed framework is based on the observation that, at time t, the amount of evolution of a given data stream is directly proportional to the sum of all the changes in the estimator's coefficients. Note here that, in the specific context of RWDE, each of these coefficients is related to a particular spatial location and hence changes in data concentrations at given locations directly produce variations in the value of the corresponding coefficients. In this sense, a high level of evolution in a given data stream is associated with significant changes over time in the estimator's coefficients.

An important advantage of the proposed framework is that since velocity densities are separately computed for each dimension then evolution coefficients can also also be obtained in the same fashion. For the case of highly dimensional data, this is an important capability that allows the straightforward identification of dimensions in data in which data evolution is relevant and dimensions in which data is not significatively changing. In this way the LEC corresponding to the dimension d of the data is defined by:

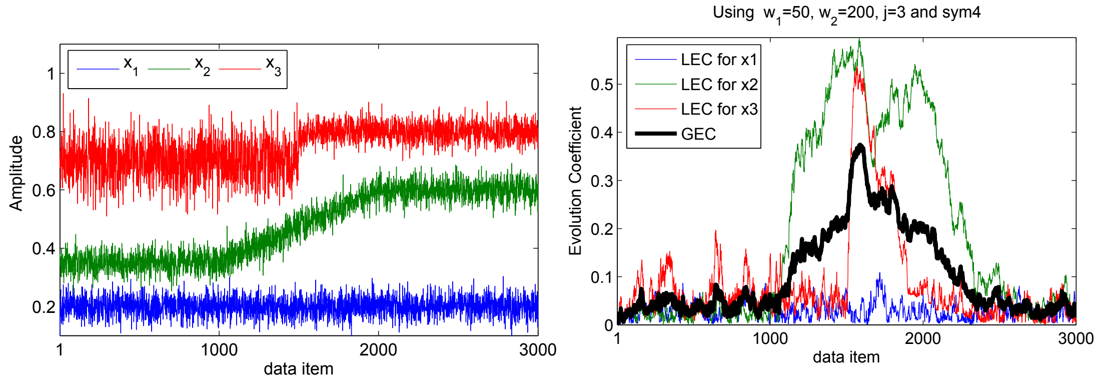
$$E^d_{(\theta_{w_1},\theta_{w_2})}(t) = (w_2 - w_1)\sum_k |\hat{V}^d_{(\theta_{w_1},\theta_{w_2})}(k,t)| \tag{5.16}$$

Note here that by using Equation (5.16) not only it can be quantified in which particular dimension the data are more significantly evolving but also it can be found a subset of the most relevant dimensions for that underlying data change.

If a general overview of the evolution is needed then a global evolution coefficient can be used. In this work it is proposed to estimate the GEC by averaging the evolution coefficients at each dimension $E^d_{(\theta_{w_1},\theta_{w_2})}(t)$ according to:

$$\tilde{E}_{(\theta_{w_1},\theta_{w_2})}(t) = \frac{1}{m}\sum_{d=1}^{m} E^d_{(\theta_{w_1},\theta_{w_2})}(t) \tag{5.17}$$

The corresponding LECs and GEC for an example 3-dimensional data stream are shown in Figure 5.4. It can be noted from Figure 5.4(b) that the LECs that detect the most relevant changes are the ones associated with the dimensions in which the data is changing, that is x_2 and x_3. Regarding the GEC, Figure 5.4(b) shows that the global evolution for the example data stream significantly increases around the time stamps 1000 and 1500, which are the time stamps related to the changes in x_2 and x_3.



(a) (b)

FIGURE 5.4: Evolution Coefficients. (a) A three dimensional example data stream; (b) the corresponding LECs and GEC.

5.3.3.2 Correlation Between Evolution Coefficients

Since the proposed framework is based on the estimation of evolution separately at each dimension then, the correlation structure of the data can be assessed by finding the correlation between evolution coefficients corresponding to different dimensions. In this way, the different degree of correlation between changes related to different dimensions can be found. To this end, it is proposed to compute the correlation coefficient (CC) between two one-dimensional streams using the Exponential Weighted Moving Correlation Coefficient

(EWMCC) algorithm of Equation (5.18). Note that the EWMCC algorithm is in turn based on an exponential discounting concepts. For simplicity let us consider a_t and b_t to be the data items at time stamp t for the data streams a and b, respectively. Then the proposed EWMCC algorithm between streams a and b can be expressed as:

$$EWMCC_{(t)}(a,b) = \frac{C_t - N_t\bar{A}_t\bar{B}_t}{\sqrt{D_t^2 - N_t(\bar{A}_t)^2}\sqrt{E_t^2 - N_t(\bar{B}_t)^2}} \tag{5.18}$$

with:

$\bar{A}_t = (1 - {N_t}^{-1})\bar{A}_{t-1} + {N_t}^{-1}a_t$; $\quad C_t = (1 - \frac{1}{w_\alpha})C_{t-1} + a_t b_t$; $\quad E_t = (1 - \frac{1}{w_\alpha})E_{t-1} + b_t^2$;

$\bar{B}_t = (1 - {N_t}^{-1})\bar{B}_{t-1} + {N_t}^{-1}b_t$; $\quad D_t = (1 - \frac{1}{w_\alpha})D_{t-1} + a_t^2$; $\quad N_t = (1 - \frac{1}{w_\alpha})N_{t-1} + 1$.

where the user defined term w_α denotes the window size, with $0 \leq \frac{1}{w_\alpha} \leq 1$.

Note that we have inserted the temporal index to the correlation since it is a recursive and time dependant variable. Note also that all the elements of Equation (5.18) are also computed recursively. Here $\bar{A}_t$ and $\bar{B}_t$ are the recursive version of the mean of a_t and b_t, respectively. Similarly, D_t and E_t are the recursive versions of the sum of a_t^2 and b_t^2, respectively. The recursive implementation of the product of a_t and b_t is given by C_t; whereas N_t is a recursive variable related to the number of data items effectively included inside the exponential window. It is also important to mention that for the computation of $\bar{A}_t$ and $\bar{B}_t$ we use the term ${N_t}^{-1}$ instead of $\frac{1}{w_\alpha}$ to provide an improved approximation for time stamps in which the number of past data samples is lower than the size of the window w_α, that is when $t < w_\alpha$. For initialisation, it is considered that $\bar{A}_1 = a_1$, $\bar{B}_1 = b_1$, $C_1 = a_1 b_1$, $D_1 = a_1^2$, $E_1 = b_1^2$ and $D_1 = 1$. The pseudo code for the proposed EWMCC is shown in Algorithm 5.3.

5.3.3.3 Finding Relevant 2D Projections of Data

The plotting of TVPs and SPVs is relevant for all those pair of dimensions (2D projections) in which the correlation structure of the data is more significantly evolving. Since the computational complexity of the recursive calculation of CCs is minimal, the procedure to find relevant pairs of dimensions involves estimating the CCs among all combinations of pairs of dimensions. At first glance, this procedure may seem prohibitive, however by noting that some of the recursive variables involved in Equation (5.18) are repeatedly used for each pair of dimensions then its applicability becomes feasible. Specifically, by

Algorithm 5.3: EWMCC(a_t,b_t,θ,γ_{t-1});

```
Input: $a_t$: Data item at time $t$ for the data stream $a$; $b_t$: Data item at time $t$ for the data stream
        $b$; $w_\alpha$: Window size parameter; $\gamma_{t-1} = \{\bar{A}_{t-1},\bar{B}_{t-1}, C_{t-1},D_{t-1},E_{t-1}, N_{t-1}\}$: Set of
        recursive parameters for the estimation of the correlation coefficient.
Output: $EWMCC_{(t)}(a,b)$: Correlation coefficient between streams $a$ and $b$ at time $t$; $\gamma(t)$:
        Updated set of recursive parameters.
$\bar{A}_t = (1 - {N_t}^{-1})\bar{A}_{t-1} + {N_t}^{-1}a_t$;
$\bar{B}_t = (1 - {N_t}^{-1})\bar{B}_{t-1} + {N_t}^{-1}b_t$;
$C_t = (1 - \frac{1}{w_\alpha})C_{t-1} + a_t b_t$;
$D_t = (1 - \frac{1}{w_\alpha})D_{t-1} + a_t^2$;
$E_t = (1 - \frac{1}{w_\alpha})E_{t-1} + b_t^2$;
$N_t = (1 - \frac{1}{w_\alpha})N_{t-1} + 1$.
$EWMCC_{(t)}(a,b) = \frac{C_t - N_t \bar{A}_t \bar{B}_t}{\sqrt{D_t^2 - N_t(\bar{A}_t)^2}\sqrt{E_t^2 - N_t(\bar{B}_t)^2}}$
comment: Evaluate the EWMCC between the one-dimensional streams of evolution coefficients
$a$ and $b$ at time $t$.
```

considering that in a m-dimensional data stream there are N_c 2-combinations of the form $\binom{m}{2}$, then to compute all of them using Equation (5.18) we require to keep m recursive means (either $\bar{A}(t)$ or $\bar{B}(t)$), m recursive sums of squared one-dimensional data streams (either $a(t)^2$ or $b(t)^2$) as well as N_c recursive products $a(t)b(t)$. Note that we only require to keep one recursive variable $D(t)$ since it is common to all dimensions. Then, in total $2m + N_c + 1$ recursive variables need to be maintained. For instance if $m = 100$ we have $N_c = 4925$ combinations of pairs of dimensions for which the storage of 5951 variables is required. Furthermore, the updating of each of these recursive variables involves only two simple multiplications.

5.3.3.4 Computational Complexity

The complexities of the proposed VDD and VDC are directly related to the complexity of the one-dimensional RWDE in which they both rely on. RWDE in turn involves, for each data item X_i, the evaluation of the scaling function $\phi_{j_0,k}(X_i)$ for each of the $N_b = (2^{j_0} + 2n_\phi - 2)$ scaling functions employed using j_0. The evaluation of $\phi_{j_0,k}(X_i)$ is performed using the so called Daubechies-Lagarias Algorithm (see [74] for more details) where two are the variables involved, namely, the order of the filter n_ϕ and the precision of the algorithm r (this work considers $r = 9$). Specifically, the complexity of evaluating $\phi_{j_0,k}(X_i)$ for a single scaling function is $O(r(2n_\phi - 1)^3)$. In this sense, $O(rN_b(2n_\phi - 1)^3)$ is the complexity of updating all the estimator's coefficients at every time stamp. According to this, the complexity of the one-dimensional VDC is $O(2rN_b(2n_\phi - 1)^3)$ while the complexity of the

m-dimensional VDC is $O(2mrN_b(2n_\phi - 1)^3)$. Since VDD, in addition to the updating of the estimator's coefficients also involves the evaluation of the density in a set of P points, its complexity for the one-dimensional case is given by $O(2r(N_b + P)(2n_\phi - 1)^3)$ whereas for the m-dimensional it is $O(2mr(N_b + P)(2n_\phi - 1)^3)$.

5.3.3.5 Computational Complexity Comparison

In this section, the computational complexity of the proposed VDD and VDC methods is theoretically compared with the velocity density framework proposed in [9], which from now on is referred to as VDKDE. For the one-dimensional case, the complexity of VDKDE is $O(2h_tP)$, where h_t is the length of the sliding window and P is the number of points in which the density is evaluated. For the m-dimensional case it is $O(2mh_tP^m)$. Note here that the number of points in which the densities are evaluated depends on the number dimensions m.

For comparison, the number of multiplications required to compute velocity densities using VDC, VDD and VDKDE, varying the number of dimensions of the data m, the size of the window for the analysis h_t and the number of P points in which each density is evaluated, is shown in Table 5.1.

TABLE 5.1: Complexity comparison of velocity density methods.

m	h_t	P	**VDC***	**VDD***	**VDKDE**
1	1000	**10**	1.4e+05	2.0e+05	2.0e+04
10	1000	**10**	1.4e+06	2.0e+06	2.0e+14
50	1000	**10**	6.8e+06	9.9e+06	1.0e+55
1	5000	**10**	1.4e+05	2.0e+05	1.0e+05
10	5000	**10**	1.4e+06	2.0e+06	1.0e+15
50	5000	**10**	6.8e+06	9.9e+06	5.0e+55
1	10000	**10**	1.4e+05	2.0e+05	2.0e+05
10	10000	**10**	1.4e+06	2.0e+06	2.0e+15
50	10000	**10**	6.8e+06	9.9e+06	1.0e+56
1	1000	**20**	1.4e+05	2.6e+05	4.0e+04
10	1000	**20**	1.4e+06	2.6e+06	2.0e+17
50	1000	**20**	6.8e+06	1.3e+07	1.1e+70
1	5000	**20**	1.4e+05	2.6e+05	2.0e+05
10	5000	**20**	1.4e+06	2.6e+06	1.0e+18
50	5000	**20**	6.8e+06	1.3e+07	5.6e+70
1	10000	**20**	1.4e+05	2.6e+05	4.0e+05
10	10000	**20**	1.4e+06	2.6e+06	2.0e+18
50	10000	**20**	6.8e+06	1.3e+07	1.1e+71

*Using $r = 9$, $n_\phi = 4$, $j_0 = 4$

The most important observation from Table 5.1 is that, since the proposed VDC and VDD algorithms are independent of the window size h_t, they are significantly less complex than

VDKDE, specially for cases involving higher dimensions. If, for instance, it is required to analyse a 50-dimensional data stream using a sliding window size of 1000 and considering 10 points for the evaluation of the forward and reverse densities at each dimension, VDKDE will require, at every time step, 1.0e+56 multiplications, while the proposed VDC and VDC will require 6.8e+06 and 9.9e+06, respectively.

5.4 Empirical Evaluation

In this section the empirical evaluation of the proposed data stream evolution diagnosis framework is presented[2]. The evaluation consists of three parts. The first one is the assessment for the selection of parameters of the algorithms. The second part includes performance comparisons against the VDKDE which is selected as benchmark algorithm. Finally, the third part includes the application of the framework in two real world applications.

5.4.1 Assessment of the proposed algorithm

The assessment of the proposed framework includes comparisons between evolution coefficients obtained varying one of the parameters of the proposed algorithms at a time namely, the order of the scaling function n_ϕ, the family of the scaling function, the initial resolution 2^{-j_0}, the pair of window sizes for the analysis w_1 and w_2. Note that comparisons involve evolution coefficients since they convey, in a single number, the total amount of variation contained in the whole velocity densities. For this assessment the one-dimensional data stream depicted in Figure 5.5a is used, which from data item 1001 to 2000 shows a representative evolution.

In Figure 5.5f it can be seen that, for the proposed framework, different wavelet families can capture the evolution of a data stream. Wavelets from the *Symlets* family are chosen as they are the "least asymmetric" compactly supported orthogonal wavelets usually selected in most of the applications involving WDE [74].

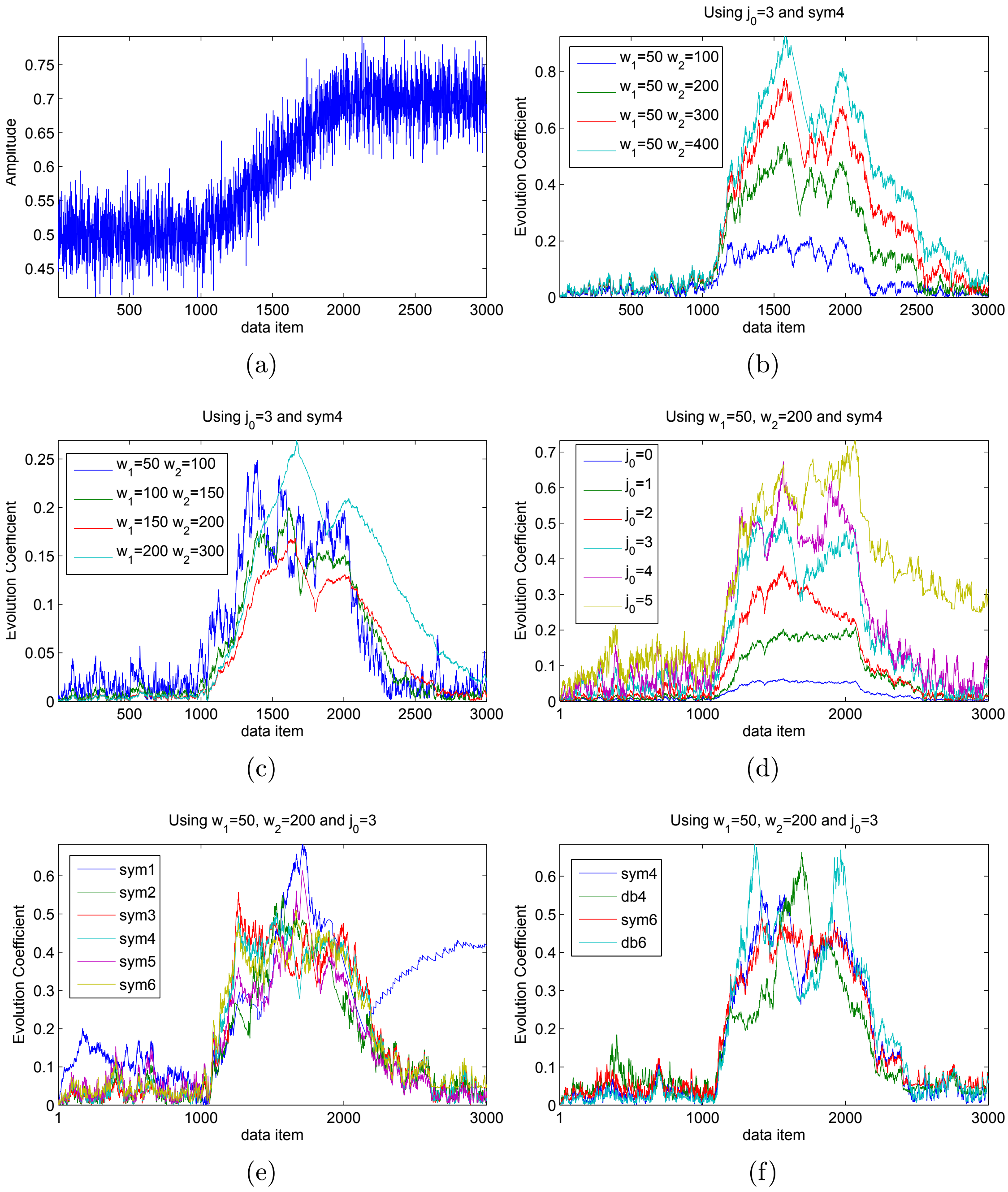
The order of the scaling function n_ϕ also impacts the resulting evolution coefficient as it is shown in Figure 5.5e. In general, it can be observed from Figure 5.5e that from the six

[2] The computer system used to generate the results reported in this section was an Intel i5 2500k with 16 GB of RAM, running on Linux Ubuntu 11.04 and the simulation environment was MATLAB R2010b.

orders evaluated only the wavelet of order 1, that is *Sym1*, presents a degraded behaviour. It is important to highlight that n_ϕ is directly related to approximation capabilities of the scaling function and in that sense, it is recommended to use $n_\phi = 4$ which is usually the order selected in the WDE literature [74].

The pair of window sizes w_1 and w_2 are application specific parameters since they define the time horizon for the analysis. Figure 5.5b and Figure 5.5c evaluate the effect of modifying the difference $w_2 - w_1$ (by maintaining w_1 fixed and only changing w_2) and the effect of simultaneously increasing w_1 and w_2 (by maintaining constant the difference $w_2 - w_1$). It can be observed from Figure 5.5b and Figure 5.5c that different pairs of window sizes are useful to distinguish the evolution in the example data stream. Note however that as the difference between w_2 and w_1 increases the magnitude of their corresponding evolution coefficients also increases. Note also that, when using larger window sizes high values for the evolution coefficients persist, some time stamps after the data stream has finished to change. According to Figure 5.5b this period of high evolution values is approximately equal to two times w_2.

Regarding the relation between evolution coefficients and the resolution 2^{-j_0}, in Figure 5.5d it can be seen that, apart from $j_0 = 0$, different values of j_0 produce evolution coefficients able to correctly detect the evolution in the example data stream. The selection of a particular value for j_0 will mainly depend on the underlying distribution of the data as well as the number of data items considered for the estimation, which, in the proposed framework, is controlled by the size of the windows w_1 and w_2. As in this case the underlying distribution of data is unknown, for the selection of j_0 it is proposed to follow a similar approach than the one used to select the bandwidth parameter in KDE, that is to select j_0 according to the optimal value for a normal distribution. Therefore, using 100 random samples of size n from a normal distribution $\mathcal{N}(0,1)$, an approximation of the Mean Integrated Squared Error (MISE) between the true density and their WDE estimates is obtained. Specifically five different values j_0 are evaluated using 500 values of n, ranging from 10 to 10000 data items. Then, for each value of n the value of j_0 that reports the lowest MISE is selected as optimal. It is important to recall that for WDE and RWDE input data requires to be normalised between the range $[0, 1]$ and that we do not consider the variance of the data for the selection of j_0. In Figure 5.6 we show the optimal values of j_0 for the different values of n evaluated.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 5.5: Evolution coefficients for an example data stream (a) versus: (b) the difference $w_2 - w_1$; (c) the pair of window sizes w_1 and w_2; (d) the index j_0 related to the resolution 2^{-j_0} ; e) the order of the scaling function n_ϕ; (f) the scaling function employed.

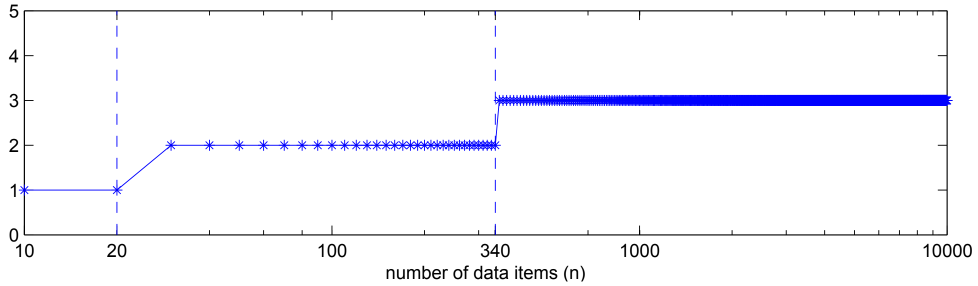


FIGURE 5.6: Selection of j_0.

5.4.1.1 Data Items Processing Capacity in the Estimation of Global Evolution Coefficients

This section focuses on the evaluation of the number of data items that can be processed per second for the estimation of the GECs. For this purpose the following variables are changed: i) the order of the filter n_ϕ, and ii) the initial resolution 2^{-j_0}. Note that since n_ϕ is similar for *Daubechies* and *Symlets* families the reported results are valid for both wavelet families.

The first observation from Table 5.2 is that the number of data items that can be processed per second is inversely proportional to the order of the filter. It can also be noted that, whichever value for n_ϕ is chosen, GECs can be updated about 10 times per second for a 100-dimensional data stream. Then, the real time estimation of GECs for highly dimensional data is clearly feasible with the proposed approach. Regarding the number of data items that can be processed per second varying the resolution 2^{-j_0}, it can be seen from Table 5.2, that in general it is of the same magnitude for different values of the index j_0.

5.4.2 Comparisons against benchmark algorithm

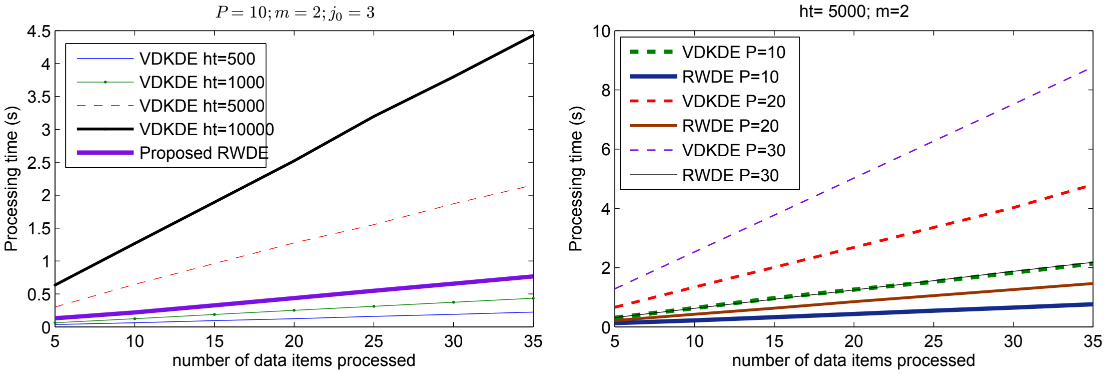
5.4.2.1 Computational complexity

In this section the time complexity of proposed and benchmark velocity density frameworks in the construction of TVPs is empirically evaluated. There are two aspects included in the comparisons: i) the number of data items processed per second versus the time window h_t; ii) the number of data items processed per second versus the number of points P chosen

TABLE 5.2: Data items per second for the estimation of GECs.

		Order of the Filter n_ϕ					
m	j_0	1	2	3	4	5	6
1	0	4522	2215	1356	963	846	818
	1	4437	2180	1361	959	843	808
	2	4225	2159	1348	957	850	808
	3	3961	2086	1313	952	842	807
	4	3568	2000	1310	957	842	799
	5	3317	1984	1299	945	818	800
10	0	501	230	139	97	86	82
	1	482	229	139	97	85	81
	2	457	225	137	98	86	82
	3	427	216	134	97	85	81
	4	382	206	134	96	85	81
	5	353	204	133	96	85	79
100	0	51	23	14	10	9	8
	1	49	23	14	10	9	8
	2	46	22	14	10	9	8
	3	43	22	13	10	8	8
	4	38	21	13	10	9	8
	5	36	21	13	10	8	8

for the evaluation of the velocity density. The corresponding results are shown in Figure 5.7.



(a) (b)

FIGURE 5.7: Number of data items processed per second: a) for different values of h_t; b) for different values of P.

Note that the time required for the proposed approach to process a particular number of data items is independent from the value of h_t and in that sense independent from the number of data items covered by the exponential window. For this reason, in Figure 5.7a only one curve is plotted for the proposed RWDE algorithm. In contrast, for the VDE method based on KDE different different values of h_t correspond to different processing times for a given number of data items. It is clear from Figure 5.7a that even though for small values of h_t the VDKDE framework requires less processing time for the construction of TVPs, in general the proposed approach is preferred since the obtention of TVPs is independent of the time horizon for the analysis.

Regarding the number of points P selected at each dimension for the evaluation of TVPs, it can be observed from Figure 5.7b that, for similar values of β, TVPs obtained with the proposed framework have a reduced processing time. For instance, in about two seconds the proposed approach is able to provide around 35 updates of the TVP using 30 points for each dimension, while the VDKDE algorithm requires about 4 seconds.

5.4.2.2 Detection Capabilities

The detection capabilities of the proposed and benchmark velocity density estimation algorithms are compared in this section. For this purpose, one stream from each subcategory of the synthetic data set of Table 4.7 from Chapter 4 (which includes streaming data from twelve subcategories of changes) is used. The experiment consists in obtaining, for each stream selected, the local evolution coefficient at each time stamp using VDKDE and VDC frameworks with different values for h_t, for the case of VDKDE, and with different combinations of window sizes w_1 and w_2, for VDC. For VDKDE the value for the spatial kernel smoothing parameter h_s is determined using the Silverman's approximation rule [62] as it is suggested by [9]. According to Silverman's approximation rule, the smoothing parameter for a data set with n points and standard deviation σ is given by $1.06\sigma n^{-1/5}$. Regarding VDC, $Sym4$ is selected as basis function since it is the usual choice for WDE [74]. In respect to j_0 it is set to 3 following the criteria suggested in Section 5.4.1. Note also that, for this particular experiment w_1 is selected to be equal to $w_2/4$. The corresponding results are depicted in Figure 5.8 to Figure 5.10.

Two are the main observations from Figure 5.8 to Figure 5.10. The first one is that the two velocity density frameworks evaluated have different detection capabilities which can be adjusted by varying their corresponding parameters. The second aspect that is important to highlight is that the benchmark approach fails to detect the evolution in the sixth to the twelfth data streams which are the ones involving changes in frequency content (data streams 7 to 9) and changes in the deterministic generation process (data streams 10 to 12). On the contrary, the proposed VDC framework is able to detect changes in the twelve types of data streams studied. The reason for this is directly related to the type of estimator employed by the proposed velocity density framework: as it has been pointed out in [186], WDEs are able to capture local features in the density since they rely on basis functions with good localisation properties in time and frequency.

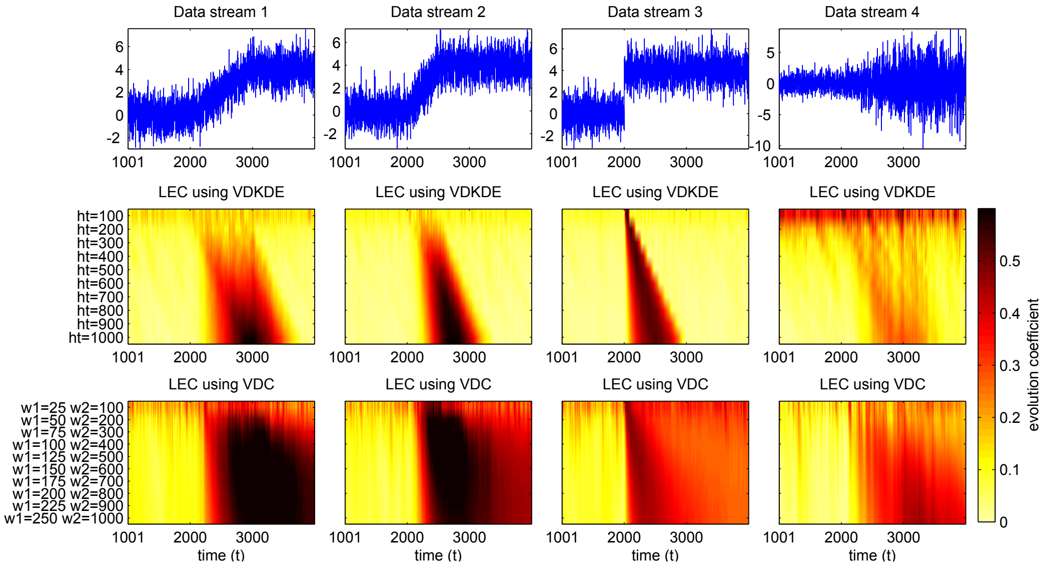


FIGURE 5.8: Data streams from the first four subcategories of changes and their corresponding LECs using VDKDE and VDC with different parameters.

Another aspect that is important to highlight from Figure 5.8 to Figure 5.10 is that among the four main types of changes studied, the most challenging one for proposed velocity density framework is the type related to changes in the frequency content of the stream (data streams 7 to 9). The reason for this is that this particular type of change does not distinctly modify the density of the stream. In order to improve the detection capabilities regarding changes in the spectrum, the stream would need to be decomposed into different frequency bands, and then apply the VDC algorithm to each band. However, this procedure will increase the computational complexity of the algorithm. Note that this is the approach followed in the OSGD-based detection framework proposed in Chapter 3 for one-dimensional data.

It is also important to point out that, as it is expected, among the three subcategories included within each of the four main type of changes, shown in Figure 5.8 to Figure 5.10, the ones involving low linear changes (data streams 1, 4, 7, and 10) are the most challenging for both the VDKDE and the VDC frameworks.

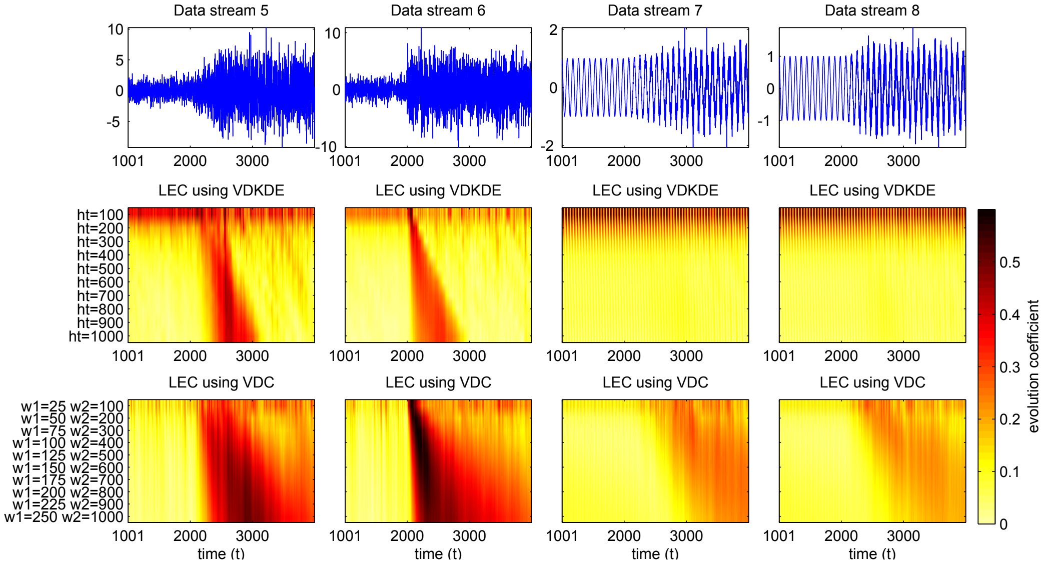


FIGURE 5.9: Data streams from the fifth to the ninth subcategory of changes and their corresponding LECs using VDKDE and VDC with different parameters.

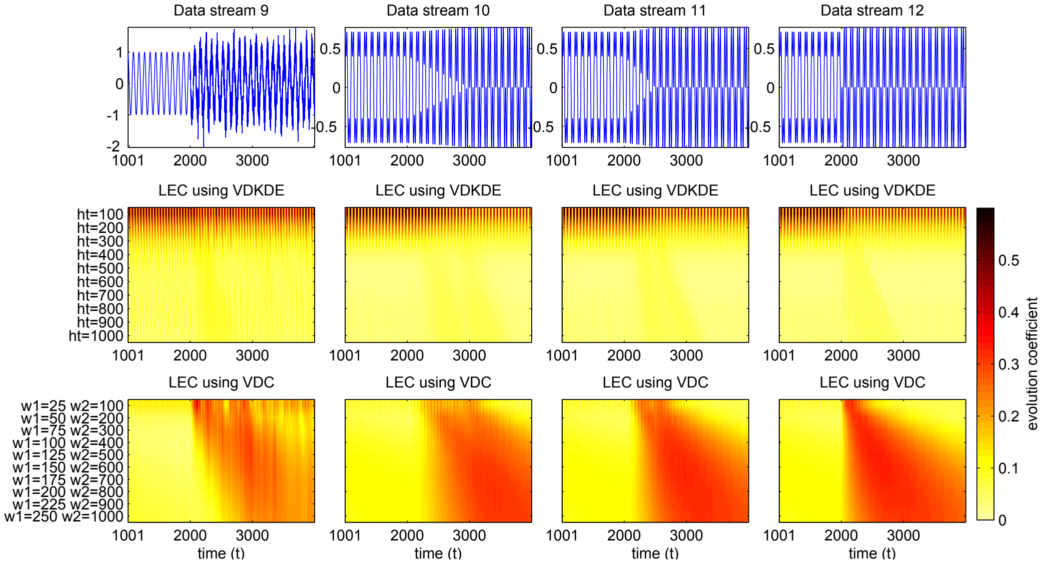


FIGURE 5.10: Data streams from the last four subcategories of changes and their corresponding LECs using VDKDE and VDC with different parameters.

5.4.3 UK's Daily Temperature from 1960-2006

In this section, the proposed data stream evolution diagnosis framework is evaluated using the daily 5 km x 5 km gridded temperature data set for the UK from the UK's National Weather Service (Met Office) [184]. The daily gridded data set of mean temperatures for the period 1960 to 2006 is used. The grid includes 10, 359 cells, identified using the Ordnance Survey National Grid, and covers the whole of the UK (including Northern Ireland). Each cell consists of 17, 167 values, where each value represents an estimate for the mean temperature of the centre point of the 5 x 5 km grid cell for a given day between 1960-2006. Each time series of mean temperatures for a given cell is treated as a single data stream and in that sense the data set includes $10,359$ streams.

The experiment consists in obtaining the VDC, and in turn the corresponding local evolution coefficient at each point in the grid for each of the $17,167$ time stamps. Two pairs of window sizes of the form $[w_1, w_2]$ are evaluated. The two pair of windows consider the period of one year for w_1, that is $w_1 = 365$. Regarding the second window, $w_2 = (5)(365) = 1825$ and $w_2 = (10)(365) = 3650$ are used for the first and second pair of windows, corresponding to five years and ten years, respectively.

In order to facilitate the visualisation of results, maps of the UK showing the average of evolution coefficients over 1 year (Figure 5.11a and Figure 5.11c) and over 5 years (Figure 5.11b and Figure 5.11d) are shown. The first observation from Figure 5.11 is that, as it is expected for the case of temperature in a particular region, results obtained are spatially correlated. It can can also be observed from Figure 5.11a and 5.11b that in general the southern part of the UK is the one with the highest temperature evolution for the two time periods studied. Another important aspect that is worth highlight is that both the Western islands and Orkney islands, in the north west and north east of the maps, respectively, also show higher temperature evolution respect to the rest of the northern areas of the country.

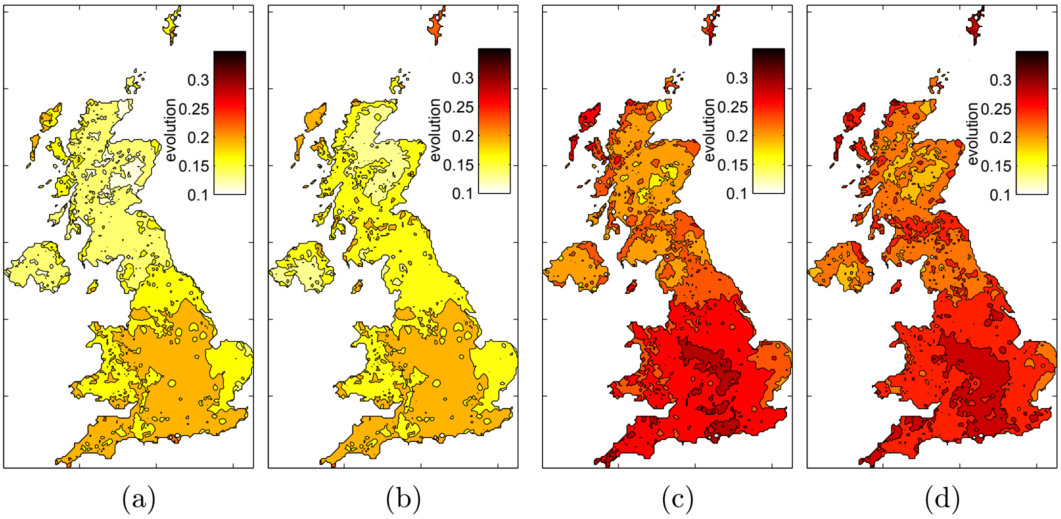
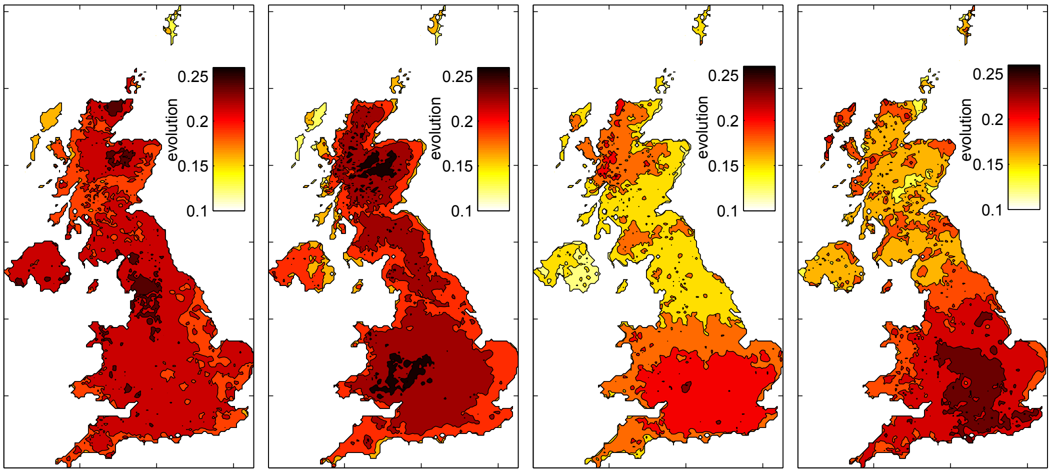


FIGURE 5.11: Averaged evolution coefficient for daily temperatures in the UK for year 2006, over 1 year (a) and over 5 years (b) using $w_1 = 365$, $w_2 = 1825$; over 1 year (c) and over 5 years (d) using $w_1 = 365$, $w_2 = 3650$.

An additional observation regarding Figure 5.11 is that the averages of evolution coefficients over 1 year and over 5 years obtained using the same pair of windows are very similar with a correlation of 93.61% between Figure 5.11a and Figure 5.11b, and a correlation of 89.04% between Figure 5.11c and Figure 5.11d. This means that the averaged temperature evolution for all the days in the year 2006 respect to the five or ten previous years was similar to the averaged evolution for all the days in the years 2002 to 2006 in respect to the same precedent years reference. Note also that results corresponding to different pair of windows are different. Particularly, it can be observed that as the difference between w_1 and w_2 increases the related evolution also increases, this is the expected result since more temperature variation is involved in larger time horizons.

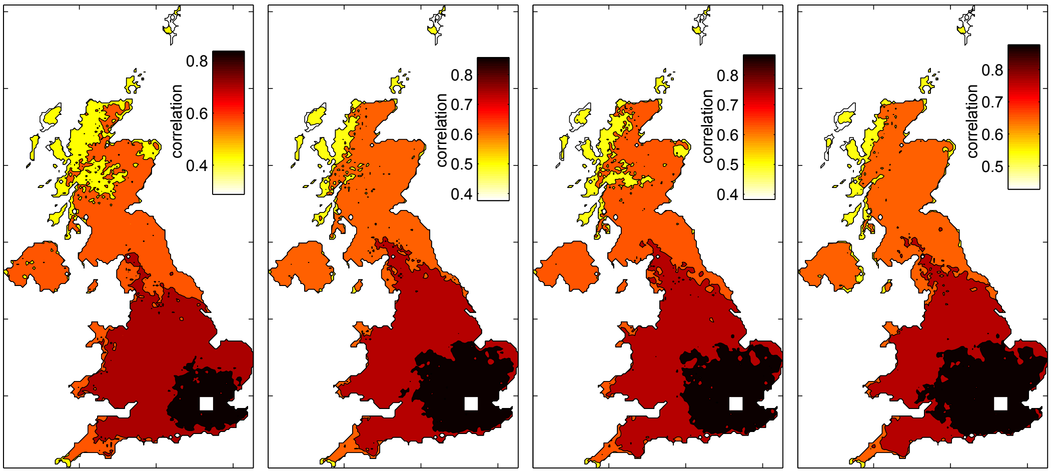
In Figure 5.12 the annual averages of evolution coefficients corresponding to years 1976, 1986, 1996 and 2006, obtained using the pair of windows $[365, 3650]$, are depicted. These maps indicate that for the first two decades studied, 1976 and 1986, the temperature evolution in the UK was largely uniform across the $10,359$ sensing locations. On the contrary, for decades 1996 and 2006 the southern part of the country was the one that presented more important changes in temperature trends.



(a) (b) (c) (d)

FIGURE 5.12: Annual averages of evolution coefficients for daily temperatures in the UK for year 1976 (a), 1986 (b), 1996 (c) and 2006 (d), using $w_1 = 365$ and $w_2 = 3650$.

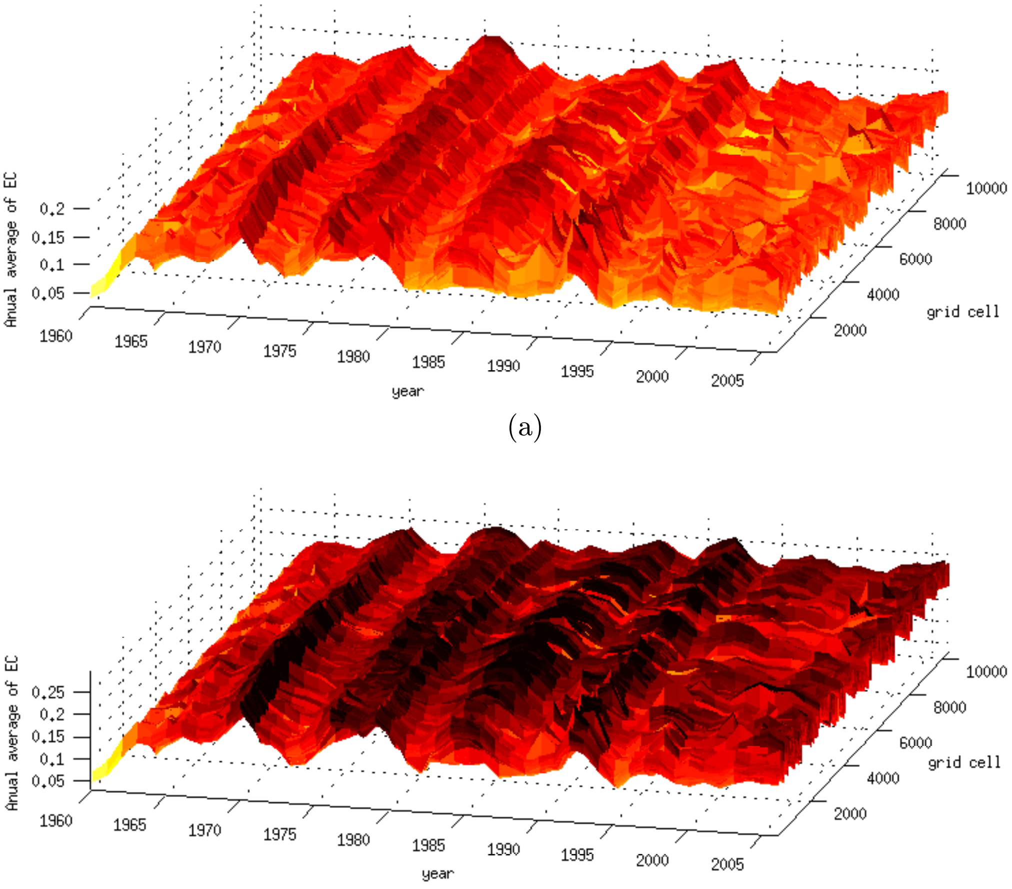
The correlation between the temperature evolution of the London borough of Westminster and the rest of the locations in the UK for the period $1960 - 2006$ is also investigated. For this purpose, at every time stamp, the correlation coefficient is computed for the past 365 ($w_\alpha = 365$) and the past 1825 ($w_\alpha = 1865$) evolution coefficients between the cell corresponding to Westminster (Easting: 530709 Northing: 179631) and the remaining $10,358$ cells in the grid. Specifically, the two pair of windows used in Figure 5.11 are used, that is $[365, 1825]$ and $[365, 3650]$. Then the resulting $17,167$ correlation coefficients at each location are averaged to produce the maps depicted in Figure 5.13. The main observation from results of Figure 5.13 is that, as it is expected for geospatial data, the temperature evolution is spatially correlated. In this sense, the remaining cell locations included in London, as well as the cells associated with East of England and South East England are the regions that report the highest correlation of temperature evolution. The second observation is related to the fact that the averaged correlation between evolution coefficients corresponding to the same pair of window sizes but with different w_α are similar, with a 2D correlation of 99.06% when using $w_\alpha = 365$ and a correlation of 98.81% for $w_\alpha = 1825$. Furthermore, the degree of similarity between results related to the same w_α but different window sizes are also similar with 99.01% correlation between Figure 5.13a and Figure 5.13b and 98.40% correlation between Figure 5.13c and Figure 5.13d.



(a) (b) (c) (d)

Figure 5.13: Correlation coefficient between evolution coefficients from 1960 – 2006 for the London borough of Westminster (white square) and the rest of the locations in the UK. (a) using $w_1 = 365$, $w_1 = 1825$ and $w_\alpha = 365$; (b) using $w_1 = 365$, $w_1 = 3650$ and $w_\alpha = 365$; (c) using $w_1 = 365$, $w_1 = 1825$ and $w_\alpha = 1825$; (d) using $w_1 = 365$, $w_1 = 3650$ and $w_\alpha = 1825$.

Surface plots related to the annual averages of evolution coefficients for the $10,359$ locations in the UK, obtained using the two pair of windows studied are shown in Figure 5.14. The main observation regarding these figures is that in both (Figure 5.14a and Figure 5.14b) four main periods of high temperature evolution across all the country can be distinguished, that is, $\sim 1965 - 1967$, $\sim 1974 - 1977$, $\sim 1982 - 1984$ and $\sim 1989 - 1991$. These periods appear when using the two pairs of windows investigated.



(a)

(b)

FIGURE 5.14: Annual average of evolution coefficients for the $10,359$ data streams of the data set using (a) $[w_1 = 365, w_2 = 1865]$ and (b) $[w_1 = 365, w_2 = 3650]$.

5.4.4 Hong Kong's Air Pollution Index for 2000-2012

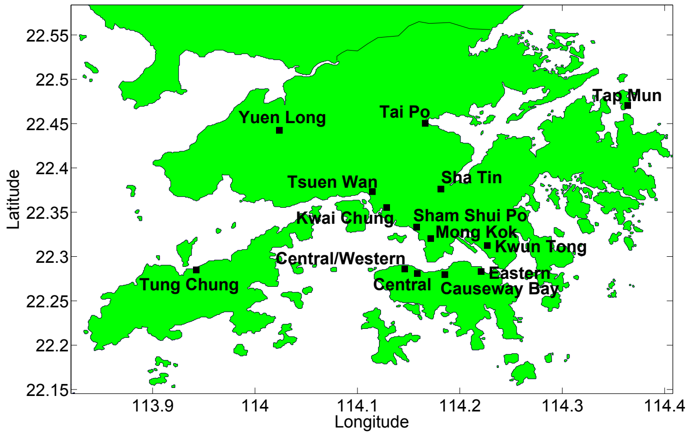
The second real world experiment considers the hourly Air Pollution Index (API) from the Air Quality Monitoring Network of the Environmental Protection Department (EPD) Hong Kong. The network comprises 14 fixed stations that monitors respirable suspended particulate (RSP), sulphur dioxide (SO2), carbon monoxide (CO), ozone (O3) and nitrogen dioxide (NO2). Figure 5.15a shows the location of the stations. Specifically, API is calculated, at each station, by first normalising the measurements of each pollutant to the scale 0 to 500 based on the 1-hour, 8-hour or 24-hour average concentrations, depending on the pollutant. Then the maximum of the normalised-averaged measurements is selected to indicate the overall pollution level at each station [187].

The experiment consists in obtaining the local evolution coefficient, for every hour between 01/01/2000 and 15/11/2012 for each of the 14 stations in the network (shown in Figure 5.15a). According to this there are 14 one-dimensional data streams of evolution coefficients with a total of $112,467$ data items each of them. One pair of window sizes is evaluated, while the first window considers the period of one month that is $w_1 = (30)(24) = 720$, the second window is set to cover the period of 5 months $w_2 = (5)(30)(24) = 3600$.

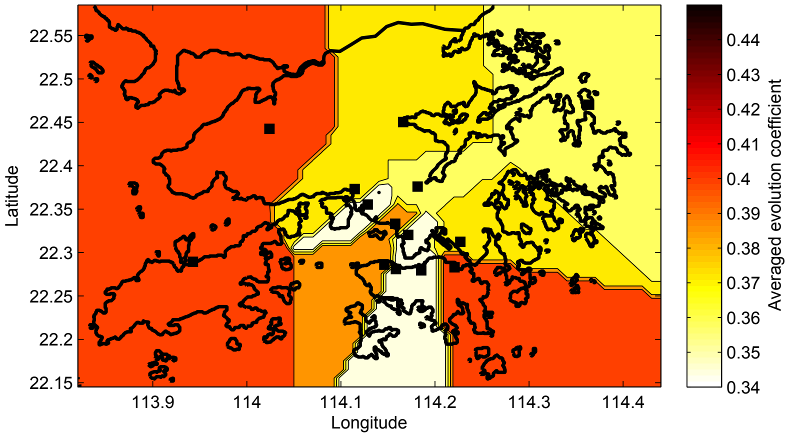
For visualisation, all the evolution coefficients related to each station are averaged and then apply a nearest neighbour interpolation to obtain a map showing the averaged evolution coefficient for the whole Hong Kong area for the period $2000-2012$. Figure 5.15b shows the corresponding results. The first observation from Figure 5.15b is that the areas covered by Yuen Long, Tung Chung and Eastern stations are the ones that report, for a time horizon of 5 months, the highest API evolution. In contrast, Kwai Chung, Mong Kok, Central and Causeway Bay are the stations whose pollution levels present the lower evolution.

In order to show how the proposed EWMCC can be useful in this type of applications, we obtain, at every time stamp, the correlation coefficient for the past 8640 evolution coefficients ($w_\alpha = 8640$, related to the number of days in a year) between each of the 91 2-combinations of the 14 stations. In this way, a data stream of $112,467$ correlation coefficients is obtained for each of the 91 combinations. Then for each combination all the resulting correlation coefficients are averaged. In this way we produce the pixel plot shown in Figure 5.16 where each pixel is related to the averaged correlation coefficient for one of the 91 combinations of the 14 stations. Note that the diagonal of the pixel plot of Figure 5.16 is related to the correlation of the evolution coefficients of a given particular station with itself and hence the value of the associated pixels is always set one.

The first important observation from Figure 5.16 is that there are two clusters in which the API evolution of the 14 stations of the network can be categorised. While the first cluster includes Causeway Bay, Central and Mong Kok stations, the second cluster comprises the remaining 11 stations. Note that, since the API evolution is more similar among stations belonging to the same cluster in Hong Kong two main zones with similar evolution in pollution levels can be distinguished, the first one is related to the central part of the city whereas the second corresponds to the surrounding areas. In contrast, it can be seen that the API of Central/Western and Tung Chung stations are highly correlated with Eastern and Yuen Long stations, respectively. According to this, when the pollution levels in the

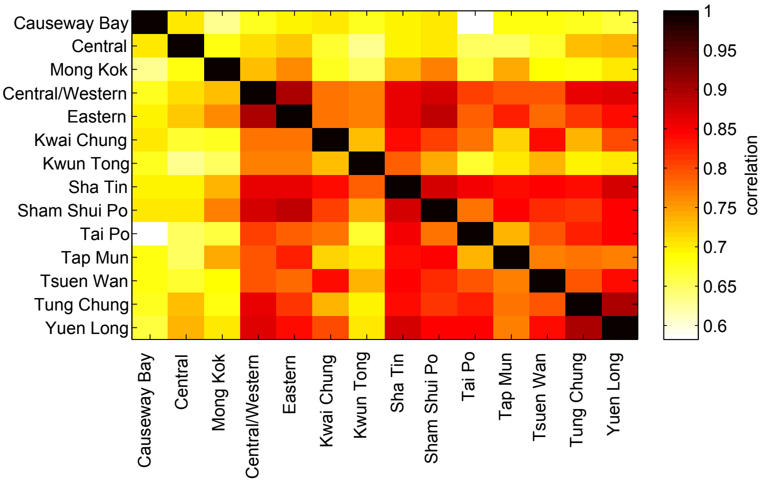


(a)



(b)

FIGURE 5.15: Results for Hong Kong's pollution experiment. (a) Location of the monitoring stations; (b) Averaged evolution coefficient using $w_1 = 720$ and $w_2 = 3600$ for $2000 - 2012$.

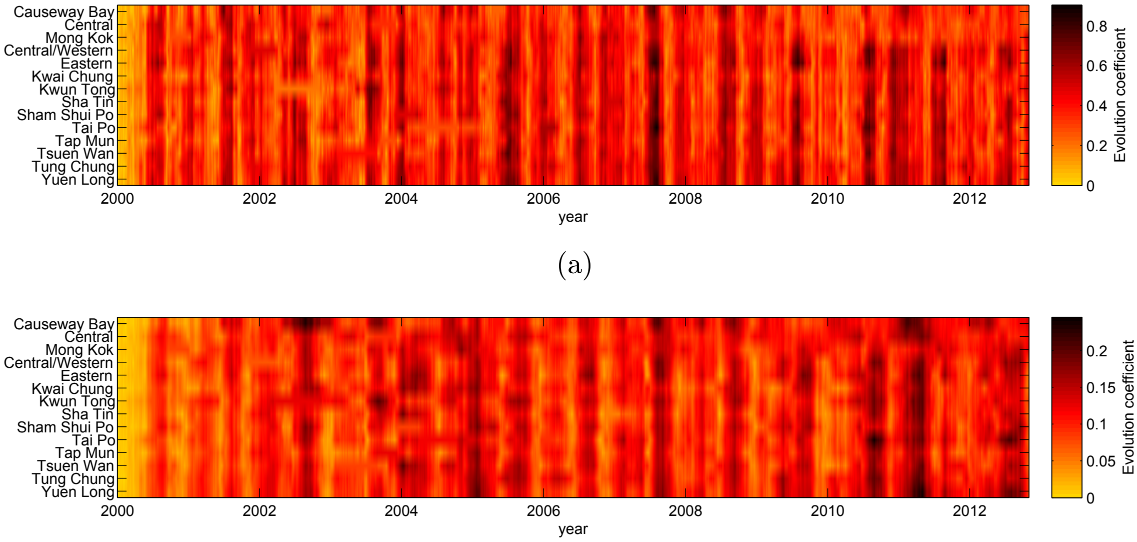


(a)

FIGURE 5.16: Averaged correlation coefficients for evolution coefficients from different stations using $w_\alpha = 3600$.

former stations present some change it can be expected that the pollution in the latter stations may also report some degree of evolution.

In Figure 5.17 pixel plots related to the local evolution coefficients for the 14 monitoring stations in Hong Kong at each of the $112,467$ time stamps are presented. While the first plot (Figure 5.17a) is related to the pair of window $[720, 3600]$, the second plot (Figure 5.17b) considers the pair $[3600, 7200]$. The most important observation regarding these figures is that there is a great amount of evolution variability for the two pair of windows evaluated. While, for Figure 5.17a the period of more evolution corresponds to the second semester of 2007, for Figure 5.17b, it is related to the first semester of 2011. The second observation is that, in general, the evolution is higher when using the first pair of windows. This means that there are larger changes in pollution levels when the evolution between the last month of API data respect to the past five months is considered; than when the evolution between the past five months of data respect to the past 10 months is considered.



(a)

(b)

Figure 5.17: LECs for the 14 stations of the API Hong Kong data base using (a) $[w_1 = 720, w_2 = 3600]$ and (b) $[w_1 = 3600, w_2 = 7200]$.

Finally, in order to show how the 2D visual profiles of Section 5.3.2 can be obtained for this experiment, the API of all monitoring stations can be considered a multidimensional data stream, where the API associated with each station is one of its 14 dimensions. Then it would be possible to think about TVPs and SVPs between different stations in the network. Figure 5.18 depicts the corresponding TVP and SVP for the 15/11/2012 considering as the first dimension data from Central station, while for the second dimension two other stations located in different areas of the city are selected, that is, Causeway Bay and Yuen Long.

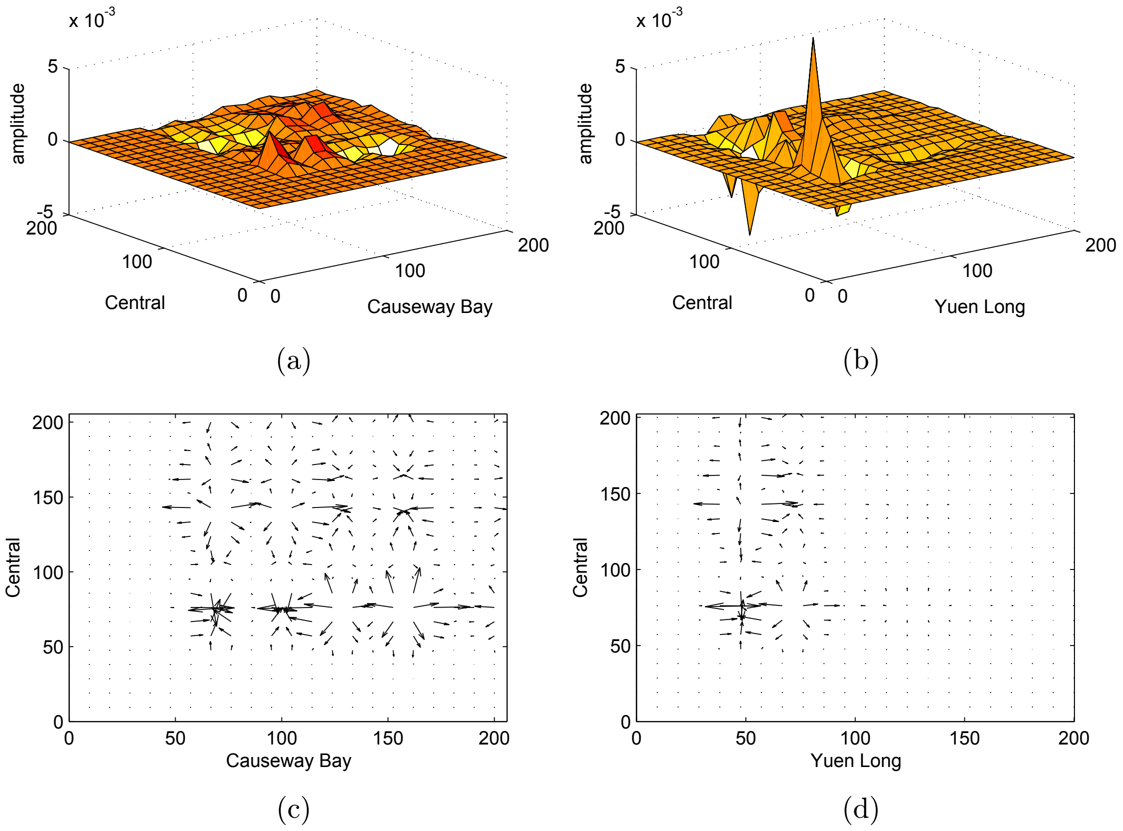


Figure 5.18: TVPs and SVPs for API Hong Kong at 15/11/2012 using $w_1 = 720$ and $w_2 = 3600$.

It can be observed from Figure 5.18 that in the TVP related to Central-Causeway Bay stations there are more locations in which the reorganisation of data takes place. This means that data is evenly changing/evolving in the two dimensions. In the corresponding SVP (Figure 5.18c) it can be seen the associated directions of these changes. Regarding TVP and SVP related to Central-Yuen Long stations, in Figure 5.18b and Figure 5.18d, it can be seen that the reorganisation of data mainly takes place on the first dimension, that is the dimension associated with the Central station. This can be explained by the fact that Yuen Long station is located relatively far from Central station and, as results from Figure 5.16 also show, there is a low correlation between the data evolution associated with this two stations.

5.5 Final Remarks

In this chapter a novel multidimensional data stream evolution diagnosis framework which extends the velocity density concepts introduced in [9] to the context of RWDE is proposed. The proposed data evolution framework provides a novel and powerful tool for the analysis of data evolution of geospatial temporal data which can complement available geospatial modelling techniques providing new insights for a better understanding of the underlying phenomena. Algorithms such as the proposed in this chapter pave the way for real time multidimensional monitoring systems.

Results reported in this chapter show the potential of the proposed framework which, for the evaluated experiments, report a significant reduction in computational complexity respect to the method proposed in [9]. Furthermore, the capability of estimating velocity densities and in turn, the capability of obtaining evolution coefficients at each dimension, makes the proposed algorithm superior and more robust compared to its KDE-based alternative.

Regarding the relation between the proposed OSGD-based detection framework of Chapter 4 and the velocity density proposed in this chapter, it can be highlighted that even though both algorithms rely on the difference of two RWDE for the assessment of the degree of change in a given data stream, the following key differences can be highlighted. The OSG-based detection framework is formulated in the context of one dimensional data streams and, since it considers a multiresolution decomposition stage, the detection of changes in a given data stream involves comparing the corresponding one-dimensional densities at each decomposition level using Euclidean distance. On the contrary, since the RWDE-based velocity density is designed to work with multidimensional data streams, the quantification of the rate of change in data involves the construction of both one-dimensional and multidimensional densities and the integration of the difference between the resulting densities.

Chapter 6

Conclusions

The aim of this thesis was to propose novel representations for time series and streaming data suitable to perform off-line and online data mining tasks. In order to achieve this aim, two main frameworks were proposed: 1) the structural generative description framework, and 2) the RWDE-based velocity density estimation framework. These frameworks and their corresponding algorithms were assessed in the context of different and diverse data mining problems in which the representation of data plays a fundamental role. The data mining tasks investigated include off-line classification of time series, change detection in data streams and online clustering of parallel data streams. This chapter summarises the research outcome and the findings of the work carried out in this thesis.

6.1 Contributions

Although the work reported in this thesis contributes to the data mining research field in a wide range of aspects, in general, the emphasis is placed on the importance of the representation stage in the process of extraction of useful information from data. The contributions made in this research can be summarised as follows:

- **The Structural Generative Descriptions (SGDs) Framework**: A novel time series representation framework was proposed, which in order to combine structural and statistical pattern recognition paradigms, moves the extraction of elemental subpatterns from data to the probability domain. This framework is based on the decomposition of time series patterns into a set of different resolution subpatterns in time or any other transformed domain. Hence, the representation strategy proposed relies on the construction fixed-length feature vectors using the sets of attributes and the sets of relations among the most elemental subpatterns obtained which are called primitives.

The main advantage of this data representation strategy is that it allows the incorporation of information about the generation process of the data at different resolutions into a compact and fixed-length feature vector. SGDs are a robust representation strategy that is able to extract the best of both worlds to create descriptions for temporal data that combine in a unique way some of the best features of structural and statistical pattern recognition paradigms. On the one hand, SGDs inherit the powerful representation capabilities of structural pattern recognition in which complex subpatterns are decomposed into simpler subpatterns, and where the structural or topological relations are taken into account for the characterisation of input patterns. On the other hand, SGDs representations express the structural generation process of the data using statistical numerical features, which not only are robust to noise and pattern distortion, but also reduce the classification task to the partition of the feature space into regions, each of them associated to a particular class.

An additional advantage of the proposed SGDs representations is the fact that they can be directly incorporated into any of the existing decision theoretic classification approach developed for static data, such as Support Vector Machines (SVMs) and Neural Networks (NNs). It is worth emphasising that, since SGDs are domain independent representations for temporal data, they are not restricted to a particular time series or data stream application.

Finally, it is important to mention that the SGDs framework can be implemented in a variety of ways since it is not restricted to the use of particular algorithms in its corresponding multiresolution decomposition and density estimation stages. A great variety of both multiresolution decomposition techniques and density estimation approaches can be combined to construct novel SGDs representations for temporal data. The key idea here is to map decomposed time series subpatterns into the probability domain by applying a density estimation operation. And then constructing fixed-length feature vectors, either using some given points sampled from the density, or using the parameters that were used to build up the density.

- **Design and implementation of Offline SGDs Algorithms**:

Two offline algorithms based on this framework as well as different alternatives for their corresponding features were proposed. While the first algorithm, SGDW, combines DTW with WDE, the second algorithm, SGDG, considers DTW and FGM. Although both algorithms are based on the same concepts, the conceptual differences regarding the estimation of the densities provide different representation capabilities. Note that while WDE is based on approximating a given density by an expansion of orthogonal functions called wavelets, FGM consider an optimisation procedure to express the density as a linear combination of Gaussian functions with different amplitudes and bandwidths.

- **Design and implementation of Online SGDs Algorithms**: In order to make the proposed SGDs representations suitable for the analysis of the increasingly relevant streaming data, the SGDs framework was also extended to the online context. For this purpose two novel online data stream representation algorithms incorporating SGDs concepts were proposed. The basic idea behind these algorithms is the online multiresolution decomposition of streaming data and the corresponding online density estimation of the resulting decomposed subpatterns. For the online multiresolution decomposition of data, in turn, two novel online decomposition methods were proposed. While the first method is based on an optimised online implementation of the DWT using a bank of scaling function filters, the second method is based on approximating the DWT decomposition using a bank of EWMA IIR filters. Regarding the online density estimation stage the two OSGDs algorithms use RWDE.

  In addition to combine statistical and structural pattern recognition paradigms, the proposed OSGDs algorithms also fulfil some of the key requirements for data stream analysis, that is, they process data in a fast and incremental way with a constant updating time and with a constant amount of memory; they provide a compact representation of each data stream at any time; and they allow both concept shift and concept drift detection.

  Regarding the compatibility with existing data stream mining techniques, it is important to mention that since the OSGDs algorithms represent data streams using fixed-length feature vectors they can be easily combined with both traditional distance measures for time series data such as DTW and Euclidean distance as well as with whichever decision theoretic-based or statistical-based algorithms such as SVMs and neural networks.

- **Design and implementation of RWDE-based Velocity Density Estimation**: A framework for diagnosing the evolution of multidimensional data streams was developed. This framework is based on the idea of incorporating a RWDE, which is a density estimation technique specifically designed for online settings, into the context of the VDE introduced in [9]. The resulting framework, which has a reduced computational complexity independent of any window size for the analysis, makes possible the fast diagnosis of data evolution at all dimensions and at relevant combinations of dimensions with only one pass of the data.

  Note that the key contribution regarding the proposed RWDE-based diagnosis framework is replacing sliding window-based *forward* and *reverse density estimates* by a pair of density estimates related to different exponential discounting/updating strategies for the coefficients of the estimators employed. This strategy not only remarkably reduces the amount memory required by the algorithm but also makes it independent of any time horizon selected for the analysis.

  An additional important contribution in this research area is the fact that, in the proposed framework, velocity densities are formulated in a separate way for each dimension, allowing localised diagnosis. Furthermore, in the proposed framework, the characterisation of changes in the structure of streaming data is based on the concept of correlation between evolution coefficients. Specifically, a recursive implementation of the Pearson's correlation coefficient based on exponential discounting was proposed.

6.2 Future Venues of Research

There are several research venues open and worth future investigation. In this section the most relevant ones are described and organised according to the corresponding framework.

- **Offline SGDs**

  Since SGDs of Chapter 3 are a new representation for time series data several venues for future research are open. Specifically, future work should explore the formulation and applicability of the proposed framework in other primary data mining tasks such as clustering, segmentation, summarisation as well as change and anomaly detection. Moreover, taking into account that the basic idea of the framework is the representation of time series by decomposing them into simpler subpatterns and the construction of generative models for each subpattern by combining generative primitives, it is clear that future work can also be directed towards the investigation of different ensembles of multiresolution decomposition and density estimation techniques that may produce improved results.

Furthermore, in Chapter 4, as a proof of concept, we evaluated the proposed SGDs framework using one of the simplest discriminative classifiers, the 1-NN algorithm, future work should investigated improvements in performance when more sophisticated methods are used in feature-based discriminative classification stage.

Taking into account the experimental evaluation of Chapter 4, in which the proposed algorithms reported outstanding results regarding motor vibration data and human ECG data, future work should be also directed towards designing specific machine condition monitoring algorithms as well as biometrics systems based on the proposed SGDs representations.

Even though WDE and FGM were suggested as methods for the density estimation block in the SGDs algorithms, the proposed framework is not restricted to a particular density estimation technique. The only requirement is a sparse density representation, which means that for the extracted subpatterns the estimated density is expressed by a reduced number of parameters or attributes. This is an interesting venue for further research, as increased discriminative power can be obtained by structural descriptions with primitives with a balanced tradeoff between sparsity and localisation.

Another interesting topic for future research is the fact that improved classification results can be expected when using SGDs representations by selecting different sets of parameters for the densities at each level of decomposition. However, note that, in order to allow the subsequent feature-based classification, all the time series in a data set need to be represented using the same set of features. Furthermore, follow up research can also be directed towards investigating procedures for the selection of points u_q since, for the proposed algorithms, better classification performance can be expected when an optimisation procedure is followed in the selection of these points.

Finally, since the proposed framework is a domain-independent approach, not restricted to particular time series or signals, future work should explore its applicability in other time series application domains.

- **Online SGDs Algorithms**

  Regarding the proposed online SGDs implementations, future work should particularly focus in the assessment of the applicability of the algorithms in other related data stream mining tasks such as classification, segmentation and motif discovery. In the context of classification, follow up research should focus on evaluating a OSGDs-based classifier in some of the typical data stream applications such as network traffic and sensor data analysis.

  Regarding segmentation, the proposed OSGDs algorithms can be adapted to break up a data stream into meaningful parts, which is a fundamental problem with many multimedia applications. In this context, the proposed algorithms can serve the important function of helping summarize mass of multimedia materials as well as providing points of access that facilitate their browsing and retrieval.

  Since the detection of repeated subsequences, also known as motif discovery, is a fundamental problem for several higher level data mining algorithms, future work can be also directed towards the formulation of online motif discovery algorithms based on the proposed OSGDs algorithms. In recent years there has been significant research effort spent on efficiently discovering motifs in time series databases using the offline approaches. However, as a result of the streaming nature of the new data that is being generated, there is the need now to perform this discovery in an online fashion. Online SGDs-based motif discovery techniques will find application on financial data assessment, robot path analysis and patient monitoring.

  In addition to this, since the empirical evaluation of OSGDs presented in Chapter 4 was more focused on showing the applicability of the algorithms in change detection and clustering problems rather than on proposing a particular algorithm, future work should also investigate data mining approaches in which more robust algorithms are employed in the distance/similarity evaluation and clustering stages. Note that, as a proof of concept, the investigation only considered the Euclidean distance combined with a simple threshold, for the case of change detection, while we only evaluate k-means, for the case of clustering.

  In regards to clustering applications, a further venue of future research is the assessment of the proposed online representations in the context of hierarchical clustering. For instance, an interesting future work could be the incorporation OSGDs into the Online Divisive Agglomerative Clustering (ODAC) approach [101].

Furthermore, since the proposed ODWT and MREWMA algorithms are by themselves promising online multiresolution decompositions with low computational cost, follow up research can also be directed towards the application of these algorithms in real-time signal processing domains. Note that, since the online multiscale decomposition of signals is extremely useful in different time series mining tasks, future work should investigate the inclusion of the proposed ODWT and MREWMA algorithms in existing data stream mining solutions.

- **RWDE-based Velocity Density Framework**

  In the context of the proposed RWDE-based VDE framework the main lines for future work include its evaluation in different application domains. Specifically, in meteorological and air pollution modelling in which is of fundamental importance to know in real time the evolution of meteorological variables or the evolution of pollutants in a given area. Note that since these two specific applications usually involve taking into account a great number of variables an evolution diagnosis technique such as the proposed RWDE-based VDE framework, which was designed to deal with online highly dimensional data, would be potentially useful.

  A third relevant application for the proposed evolution framework is the online diagnosis and prognosis of machinery health condition, where the key idea us to forecast damage propagation trend in rotary machinery and to provide alarms before a fault reaches critical levels. Note that in this context machine condition prognosis particularly refers to the use of available observations to forecast upcoming states of the machine. Prognosis is a relatively new mechanical systems and signal processing research area that is intended to complement traditional maintenance strategies commonly used in industry such as corrective and preventive maintenance.

  Biology is the fourth potential application for the proposed framework which can be used as a novel analytic tool for the assessment of phenotypic change in different plant species and organisms. Note here that rapid phenotypic changes can be caused by that environmental changes, such as climate changes.

  Finally, further research should also focus on the applicability of the proposed VDE framework in the context of robust system modelling and anomaly detection algorithms for WSN where there are important limitations regarding computational resources. In this specific context, online and multivariate diagnosis algorithms, such as the proposed one, are the key to make possible distributed sensor data analysis solutions.

Publications

[1] E. S. García-Treviño, and J. A. Barria, "Online wavelet-based density estimation for non-stationary streaming data," *Computational Statistics and Data Analysis*, vol. 56, no. 2, pp. 327-344, 2012.

[2] S. Thajchayapong, J. A. Barria, and E. S. García-Treviño. "Lane-level traffic estimations using microscopic traffic variables," *Proceedings of the13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1189-1194, 2010.

[3] S. Thajchayapong, E. S. García-Treviño, and J. A. Barria, "Distributed Classification of Traffic Anomalies Using Microscopic Traffic Variables," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, p. 448-458, 2013.

[4] E. S. García-Treviño, and J. A. Barria, "Structural generative descriptions for time series classification," *IEEE Transactions on Cybernetics*, (Submitted).

[5] E. S. García-Treviño, and J. A. Barria. "ECG-based biometric recognition framework based on structural generative descriptions," *IEEE Transaction on Information Forensics and Security*, (Submitted).

Appendix A

A.1 Daubechies-Lagarias Algorithm

In the context of WDEs and RWDEs, it is necessary to find values of compactly supported orthogonal wavelets at arbitrary points. Since scaling and wavelet functions for this type of wavelet families have no explicit representation (except for the $db1$ wavelet), then the Daubechies-Lagarias algorithm [75] is required. This algorithm provides, for all compactly supported orthogonal wavelet filters, a fast numerical calculation of the value of scaling ϕ and wavelet ψ functions at a given point with some predefined accuracy r.

Note that although the Daubechies-Lagarias algorithm is useful for all orthogonal wavelet filters, the formulation presented in this section is useful for scaling and wavelet functions from the Daubechies and Symlets families, whose support, for both ϕ and ψ functions, is $[0, 2n_\phi - 1]$ with n_ϕ denoting the order of the scaling function filter.

Let x to be an arbitrary point within the interval $(0, 1)$, and let the operator $dyad(x) = \{d_1, d_2, \ldots, d_r, \ldots\}$ denote the set of 0-1 digits in the dyadic representation of x, in such a way that $x = \sum_{j=1}^{\infty} d_j 2^{-j}$. Using $dyad(x, n) = \{d_1, d_2, \ldots, d_r\}$ to denote the subset of the r digits from $dyad(x)$.

Let $h = (h_0, h_1, \ldots, h_{2n_\phi - 1})$ be the vector of coefficients of the scaling function filter ϕ. Denoting two $(2n_\phi - 1) \times (2n_\phi - 1)$ matrices of the form:

$$T_0 = \left(\sqrt{2}h_{(2i-j-1)}\right) \quad \text{for} \quad 1 \leq i, j \leq 2n_\phi - 1 \tag{A.1}$$

$$T_1 = \left(\sqrt{2}h_{(2i-j)}\right) \quad \text{for} \quad 1 \leq i, j \leq 2n_\phi - 1 \tag{A.2}$$

In [75], Daubechies and Lagarias found that $\phi(x)$ can be approximated by considering the following expression:

$$\lim_{r\to\infty} T_{d_1}T_{d_1}\ldots T_{d_r} = \begin{bmatrix} \phi(x) & \phi(x) & \ldots & \phi(x) \\ \phi(x+1) & \phi(x+1) & \ldots & \phi(x+1) \\ \vdots & \vdots & & \vdots \\ \phi(x+2n_\phi-2) & \phi(x+2n_\phi-2) & \ldots & \phi(x+2n_\phi-2) \end{bmatrix} \tag{A.3}$$

Regarding the evaluation of wavelet function ψ at some specific point x, the procedure involves the following vector:

$$u(x) = \left\{(-1)^{1-[2x]}h_{i+1-[2x]}\right\}_{i=0,\ldots,2n_\phi-2} \tag{A.4}$$

where the operator $[x]$ represents the highest integer less than x. If for some i the index $i+1-2[x]$ is negative or larger than $n_\phi - 1$ then the corresponding element of u is zero. Let $e = (1, 1, , \ldots, 1)$ be a row vector of ones and v be defined as:

$$v(x,r) = \frac{1}{2n_\phi - 1}e^T \prod_{i\in dyad(2x,r)} T_i \tag{A.5}$$

By considering Equation A.4 and Equation A.5 $\psi(x)$ can be expressed as:

$$\psi(x) = \lim_{n\to\infty} u(x)^T v(x,r) \tag{A.6}$$

A.2 Detailed Results from Chapter 4

In this section detailed results for the four off-line experiments of Chapter 3 are presented. For the first three experiments, that is, for the Synthetic data experiment (Section 4.1.2.1), the CWRU experiment (Section 4.1.2.2) and the ECG biometrics experiment (Section 4.1.2.3) results are presented separately for benchmark and for the proposed algorithms. Table A.1, Table A.3 and Table A.5 report results for benchmark representations while Table A.2, Table A.4 and Table A.6 present results for the proposed SGDs-based algorithms. Note that the detailed results presented in these tables include the average, the minimum, and maximum of the classification error over the 100 trials, as well as the value of the corresponding tuning parameter that reported the best averaged classification error for each representation. These results are arranged according to the distance measure employed. Regarding the proposed algorithm, Table A.2, Table A.4 and Table A.6 additionally categorise the results into nine categories according to the wavelet W used by the algorithms.

Regarding the UCR experiment of Section 4.1.2.4, Table A.7 we report the performance results obtained when using ED, SE and CO as distance measure for the 1-NN algorithm, while Table A.8 include the corresponding results for CH and CR measures. This two tables include, the average classification error over 100 trials for the best time series representation (either a benchmark technique or a proposed SGD-based algorithm) for each data set, indicating within brackets the name of corresponding representation as well as the values of the tuning parameters that reported this performance.

Table A.1: Classification error over 100 trials for benchmark representations in Synthetic data experiment

d	Method	Avg	Min	Max	Node	d	Method	Avg	Min	Max	Node	d	Method	Avg	Min	Max	Node
ED	SM	12.25	10.76	13.65	(1)	SE	SM	12.33	10.62	14.13	(1)	CO	SM	44.87	42.05	47.08	(1)
	DWT	1.62	0.66	3.06	(3)		DWT	1.87	0.66	3.12	(3)		DWT	3.38	2.12	5.03	(4)
	$CHEB$	0.60	0.17	1.49	(20)		$CHEB$	0.64	0.31	1.25	(20)		$CHEB$	3.64	2.33	5.28	(18)
	PAA	0.78	0.35	1.53	(13)		PAA	0.74	0.24	1.22	(13)		PAA	2.03	1.08	3.30	(13)
	$ARMA$	26.34	24.13	28.78	(2,2)		$ARMA$	96.88	96.88	96.88	(1,1)		$ARMA$	26.81	24.34	29.38	(2,2)
	$ARIMA$	36.81	34.76	38.82	(2,1)		$ARIMA$	96.88	96.88	96.88	(1,1)		$ARIMA$	36.07	33.99	38.44	(1,2)
	MSE	1.52	0.56	2.71	(15)		MSE	1.53	0.76	2.78	(15)		MSE	14.19	11.56	17.12	(15)
	$FMSE$	8.49	6.53	11.39	(5)		$FMSE$	6.40	3.99	8.30	(15)		$FMSE$	17.40	14.62	20.38	(15)
	DFT	1.84	1.18	3.06	(30)		DFT	1.02	0.38	1.98	(30)		DFT	1.75	0.97	2.81	(30)
	$DFTW$	0.26	0.00	0.76	(15,4)		$DFTW$	0.35	0.00	0.94	(19,8)		$DFTW$	**0.37**	0.03	0.90	(13,4)
	ACF	1.19	0.24	2.08	(15)		ACF	1.59	0.76	3.19	(15)		ACF	0.93	0.24	3.12	(15)
	$POLY$	14.67	12.74	17.33	(2)		$POLY$	12.29	10.21	15.69	(2)		$POLY$	16.74	14.37	19.20	(3)
	SVD	1.64	0.49	3.72	(20)		SVD	1.57	0.59	2.85	(20)		SVD	2.76	1.63	4.03	(7)
	PCA	1.75	0.66	3.33	(20)		PCA	1.72	0.76	3.61	(18)		PCA	3.82	2.22	5.14	(19)
	WPE	11.04	8.40	13.06	(6)		WPE	25.48	22.78	31.49	(6)		WPE	10.40	8.19	13.47	(6)
	WPS	2.22	1.08	3.58	(6)		WPS	24.30	21.42	28.12	(1)		WPS	3.38	1.98	4.76	(6)
	$DWTS$	5.01	3.12	7.08	(6)		$DWTS$	11.89	9.58	15.21	(6)		$DWTS$	7.99	5.42	10.38	(6)
	DCT	1.21	0.38	2.15	(0.15)		DCT	96.88	96.88	96.88	(0.1)		DCT	2.82	1.67	4.58	(0.5)
	$DCT2$	0.81	0.24	1.63	(70)		$DCT2$	0.81	0.38	1.70	(50)		$DCT2$	2.10	1.28	3.06	(80)
	$DFT2$	**0.16**	0.03	0.80	(70)		$DFT2$	**0.20**	0.07	0.49	(80)		$DFT2$	0.44	0.14	1.32	(50)
	RAW	1.27	0.66	2.64	(1)		RAW	1.27	0.42	2.19	(1)		RAW	3.49	2.47	4.65	(1)
CH	SM	12.49	11.15	14.90	(1)	CR	SM	93.74	93.12	93.82	(1)						
	DWT	3.01	1.77	4.69	(3)		DWT	3.44	1.91	4.79	(4)						
	$CHEB$	0.79	0.17	1.70	(18)		$CHEB$	4.06	2.50	5.83	(19)						
	PAA	0.86	0.35	2.08	(13)		PAA	4.65	2.99	6.08	(7)						
	$ARMA$	27.00	24.58	29.86	(2,2)		$ARMA$	28.37	25.76	30.87	(2,2)						
	$ARIMA$	35.88	33.78	38.75	(2,1)		$ARIMA$	36.86	34.58	39.79	(1,2)						
	MSE	3.77	2.22	5.87	(15)		MSE	24.24	21.74	27.60	(15)						
	$FMSE$	10.22	7.99	12.78	(5)		$FMSE$	24.71	22.08	29.24	(15)						
	DFT	2.63	1.84	3.85	(30)		DFT	1.81	1.04	3.37	(30)						
	$DFTW$	0.27	0.00	0.52	(13,4)		$DFTW$	0.47	0.03	1.28	(19,4)						
	ACF	0.92	0.28	1.98	(15)		ACF	28.66	26.60	31.42	(15)						
	$POLY$	15.82	13.47	18.37	(2)		$POLY$	26.66	24.27	29.34	(4)						
	SVD	2.15	1.08	3.68	(10)		SVD	3.23	1.94	5.24	(14)						
	PCA	2.77	1.67	4.34	(20)		PCA	3.87	2.64	5.66	(18)						
	WPE	12.46	9.17	15.69	(6)		WPE	10.47	8.09	13.61	(6)						
	WPS	4.70	3.33	6.35	(6)		WPS	3.47	1.84	5.24	(6)						
	$DWTS$	7.11	5.21	9.72	(6)		$DWTS$	8.06	5.66	10.21	(6)						
	DCT	1.85	0.80	2.88	(0.3)		DCT	2.86	1.53	4.44	(0.5)						
	$DCT2$	1.11	0.42	2.15	(90)		$DCT2$	2.03	0.73	3.78	(70)						
	$DFT2$	**0.22**	0.03	0.90	(80)		$DFT2$	**0.42**	0.17	1.11	(60)						
	RAW	21.53	20.03	22.88	(1)		RAW	4.27	3.12	5.45	(1)						

Table A.2: Averaged classification error over 100 trials for SGD-based algorithms in the Synthetic data experiment

		SGDW1				SGDW2				SGDW3				SGDG1				SGDG2			
d	W	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node
ED	$db1$	1.81	0.31	3.12	(1,4)	7.21	4.06	11.25	(2,2)	0.33	0.00	0.94	(1,5)	11.96	9.38	15.94	(0,4)	9.66	6.25	13.75	(3,2)
	$db3$	1.42	0.31	3.12	(2,4)	6.51	3.44	10.00	(2,2)	0.46	0.00	1.56	(1,5)	12.22	8.12	17.50	(0,4)	9.23	5.62	12.81	(3,2)
	$db6$	1.44	0.31	3.44	(2,4)	9.53	5.62	13.44	(2,2)	0.77	0.00	1.88	(1,5)	12.41	8.75	16.25	(0,4)	13.35	9.69	16.56	(3,2)
	$bior1.3$	1.66	0.31	3.12	(2,4)	7.39	3.75	10.31	(2,2)	0.34	0.00	0.94	(1,5)	12.22	9.38	16.88	(0,4)	9.74	6.56	13.44	(3,2)
	$bior5.5$	1.31	0.31	3.12	(2,4)	5.03	2.81	7.19	(3,2)	0.96	0.00	1.88	(0,6)	12.16	8.44	15.94	(0,4)	15.68	11.56	19.38	(0,2)
	$coif1$	1.33	0.00	4.06	(1,5)	6.91	4.69	9.69	(2,2)	0.44	0.00	1.25	(1,5)	12.07	7.81	16.25	(0,4)	12.31	9.69	17.19	(1,2)
	$coif3$	1.64	0.31	3.44	(1,6)	9.31	5.94	13.44	(2,2)	0.55	0.00	1.88	(1,5)	11.99	8.12	16.25	(0,4)	15.58	11.56	20.00	(0,2)
	$sym2$	1.02	0.00	3.44	(2,4)	6.27	3.75	10.31	(3,2)	0.36	0.00	1.88	(1,5)	11.91	7.81	16.88	(0,4)	12.25	9.38	15.00	(3,2)
	$sym4$	1.38	0.00	3.75	(2,4)	8.24	5.00	11.25	(1,1)	0.58	0.00	1.25	(1,5)	12.24	8.75	15.94	(0,4)	15.73	12.81	19.06	(0,2)
SE	$db1$	0.49	0.00	1.88	(1,5)	13.52	10.62	17.50	(0,2)	0.29	0.00	1.56	(0,5)	5.52	3.12	9.38	(0,2)	7.83	5.31	11.25	(3,2)
	$db3$	1.72	0.62	3.12	(0,4)	13.34	9.06	17.81	(0,2)	0.30	0.00	1.56	(0,6)	5.37	2.19	8.44	(0,2)	6.71	3.75	11.25	(3,2)
	$db6$	1.68	0.62	3.12	(0,4)	13.23	9.38	17.19	(0,2)	0.31	0.00	1.25	(0,6)	5.51	2.81	10.31	(0,2)	12.89	9.38	16.88	(0,2)
	$bior1.3$	0.54	0.00	1.56	(1,5)	11.48	6.88	93.75	(1,1)	0.33	0.00	1.88	(0,6)	5.36	2.19	9.38	(0,2)	8.85	5.31	13.12	(3,2)
	$bior5.5$	1.65	0.31	3.12	(0,4)	13.61	10.31	16.88	(0,2)	0.29	0.00	1.88	(0,5)	5.59	3.12	9.38	(0,2)	12.78	9.06	15.94	(0,2)
	$coif1$	1.78	0.31	4.06	(0,4)	13.41	9.69	17.19	(0,2)	0.29	0.00	2.19	(0,6)	5.33	2.81	8.44	(0,2)	7.50	4.69	10.94	(1,2)
	$coif3$	1.70	0.31	3.44	(0,4)	13.22	9.38	18.44	(0,2)	0.31	0.00	1.56	(0,5)	5.33	2.81	9.06	(0,2)	12.97	8.75	17.81	(0,2)
	$sym2$	1.75	0.62	3.75	(0,4)	13.53	9.06	18.12	(0,2)	0.32	0.00	1.25	(0,5)	5.43	2.19	9.69	(0,2)	9.90	6.56	13.44	(2,2)
	$sym4$	1.86	0.62	3.75	(0,4)	13.30	8.75	17.19	(0,2)	0.31	0.00	1.88	(0,6)	5.39	2.81	8.44	(0,2)	12.84	9.69	16.56	(0,2)
CO	$db1$	1.56	0.31	3.44	(1,4)	7.33	4.06	11.56	(2,2)	0.27	0.00	0.94	(1,5)	12.14	8.75	15.62	(1,2)	9.28	5.31	12.50	(3,2)
	$db3$	1.70	0.62	3.75	(3,3)	6.02	3.44	10.00	(1,1)	0.62	0.00	1.56	(1,5)	12.73	9.06	16.88	(0,4)	9.21	5.62	12.19	(3,2)
	$db6$	1.87	0.00	3.44	(3,3)	11.93	7.81	15.94	(3,2)	0.95	0.00	1.88	(1,5)	12.94	10.00	15.94	(0,4)	13.18	6.88	17.50	(3,2)
	$bior1.3$	1.45	0.31	3.12	(2,4)	8.82	5.00	13.12	(1,1)	0.38	0.00	1.56	(1,5)	12.39	8.75	16.56	(1,2)	9.86	5.94	13.44	(3,2)
	$bior5.5$	1.72	0.00	4.06	(2,4)	6.56	4.06	10.00	(3,2)	1.02	0.00	2.19	(0,5)	12.81	8.12	16.88	(0,4)	15.79	11.56	20.62	(1,2)
	$coif1$	1.66	0.31	4.38	(1,5)	5.91	3.12	10.62	(1,1)	0.55	0.00	1.88	(1,5)	12.91	7.19	16.25	(0,4)	12.94	9.38	16.88	(1,2)
	$coif3$	2.15	0.31	5.00	(2,4)	10.95	7.81	14.37	(1,1)	0.72	0.00	1.88	(1,5)	12.77	8.75	17.50	(0,4)	16.05	11.56	21.25	(1,2)
	$sym2$	1.24	0.00	2.81	(2,4)	6.93	3.75	10.31	(3,2)	0.51	0.00	1.88	(1,5)	12.45	8.75	16.88	(0,4)	12.60	9.69	15.94	(3,2)
	$sym4$	2.07	0.31	5.00	(1,5)	8.58	5.62	12.19	(2,2)	0.80	0.00	2.81	(1,5)	12.69	8.44	16.88	(0,4)	16.07	12.19	20.00	(0,2)
CH	$db1$	3.51	1.56	6.25	(1,4)	9.28	6.56	12.19	(2,2)	1.32	0.31	3.12	(1,3)	13.70	10.00	17.19	(0,4)	12.46	9.06	15.62	(3,2)
	$db3$	3.07	0.94	5.00	(2,3)	9.25	5.94	13.44	(2,2)	2.87	1.25	5.94	(1,2)	13.80	9.38	18.44	(0,4)	12.23	8.12	18.12	(3,2)
	$db6$	4.31	1.88	6.56	(1,4)	11.95	8.75	16.25	(2,2)	2.88	0.94	7.50	(0,4)	13.96	10.31	20.31	(0,4)	15.73	11.56	20.00	(3,2)
	$bior1.3$	3.58	1.56	6.56	(1,4)	9.62	6.25	12.50	(2,2)	1.29	0.00	3.44	(1,3)	14.03	10.94	17.50	(0,4)	12.17	8.44	18.44	(3,2)
	$bior5.5$	3.94	2.19	5.94	(2,3)	7.24	4.69	11.88	(1,2)	2.92	0.62	5.62	(0,4)	14.07	10.31	18.12	(0,4)	17.26	12.50	21.56	(0,2)
	$coif1$	3.94	2.19	6.25	(2,3)	9.13	5.31	12.50	(1,1)	2.30	1.25	4.69	(1,2)	14.15	10.31	17.81	(0,4)	14.05	10.00	18.75	(1,2)
	$coif3$	4.24	2.19	6.56	(1,4)	12.57	9.38	16.56	(2,2)	2.76	0.31	5.94	(0,4)	13.96	9.06	18.75	(0,4)	16.85	11.88	21.56	(0,2)
	$sym2$	3.46	1.56	5.94	(2,3)	9.30	5.00	13.44	(1,1)	2.16	0.94	4.69	(1,2)	14.17	10.31	19.06	(0,4)	14.85	10.94	18.75	(3,2)
	$sym4$	3.74	1.88	6.25	(2,3)	10.13	5.00	14.69	(1,1)	2.96	0.94	5.94	(0,4)	13.80	8.44	18.12	(0,4)	16.82	12.81	20.62	(0,2)
CR	$db1$	1.61	0.31	3.75	(1,4)	7.12	3.75	10.62	(2,2)	0.31	0.00	0.94	(1,5)	12.73	8.75	16.56	(0,4)	9.48	6.56	13.44	(3,2)
	$db3$	1.66	0.31	3.44	(3,3)	5.99	2.81	8.44	(1,1)	0.64	0.00	1.88	(1,5)	12.88	9.38	17.19	(0,4)	8.83	5.00	11.88	(3,2)
	$db6$	1.77	0.31	4.38	(3,3)	12.07	8.75	16.25	(3,2)	1.01	0.00	2.81	(1,5)	13.07	9.38	16.25	(0,4)	12.94	9.06	17.19	(3,2)
	$bior1.3$	1.29	0.31	3.12	(2,4)	8.62	5.94	12.50	(2,2)	0.32	0.00	0.94	(1,5)	12.16	8.75	16.88	(1,3)	10.42	7.19	15.62	(3,2)
	$bior5.5$	1.70	0.31	3.12	(2,4)	6.38	3.44	9.38	(3,2)	1.13	0.00	2.19	(0,6)	12.84	9.06	15.94	(0,4)	15.70	11.56	19.69	(1,2)
	$coif1$	1.57	0.31	3.44	(1,5)	6.15	3.44	10.00	(1,1)	0.62	0.00	1.56	(1,5)	13.13	9.69	17.50	(0,4)	13.29	8.75	16.88	(3,2)
	$coif3$	2.23	0.94	5.00	(2,4)	10.79	7.50	15.00	(1,1)	0.79	0.00	2.50	(1,5)	12.91	9.06	17.19	(0,4)	16.48	10.31	21.56	(1,2)
	$sym2$	1.26	0.00	2.50	(2,4)	6.74	4.69	9.69	(3,2)	0.55	0.00	1.56	(1,5)	12.65	8.75	16.56	(0,4)	12.53	9.69	17.50	(3,2)
	$sym4$	2.10	0.62	4.06	(2,4)	8.82	5.94	13.75	(2,2)	0.68	0.00	1.88	(1,5)	13.06	9.38	18.12	(0,4)	15.93	11.88	20.00	(2,2)

Table A.3: Classification error over 100 trials for benchmark representations in CWRU experiment

d	Method	Avg	Min	Max	Node	d	Method	Avg	Min	Max	Node	d	Method	Avg	Min	Max	Node
ED	SM	3.28	1.06	5.29	(1)	SE	SM	13.87	10.32	19.05	(1)	CO	SM	35.49	30.16	40.21	(1)
	DWT	66.15	62.17	70.63	(6)		DWT	46.05	41.80	50.79	(1)		DWT	44.79	41.01	48.68	(6)
	$CHEB$	42.15	38.36	46.03	(15)		$CHEB$	43.57	39.15	49.21	(3)		$CHEB$	54.42	49.21	59.79	(17)
	PAA	50.08	45.50	55.56	(4)		PAA	50.24	45.24	56.35	(4)		PAA	66.76	61.90	70.63	(20)
	$ARMA$	1.07	0.00	2.91	(1,6)		$ARMA$	84.39	84.39	84.39	(1,1)		$ARMA$	0.85	0.00	2.12	(2,5)
	$ARIMA$	0.70	0.00	2.38	(4,6)		$ARIMA$	84.39	84.39	84.39	(1,1)		$ARIMA$	0.78	0.00	2.65	(2,6)
	MSE	2.75	1.32	4.76	(15)		MSE	2.57	1.06	3.97	(15)		MSE	6.12	3.70	8.47	(15)
	$FMSE$	27.70	23.54	32.54	(14)		$FMSE$	27.84	24.07	31.48	(14)		$FMSE$	31.92	28.31	36.24	(15)
	DFT	33.01	27.78	38.36	(29)		DFT	33.66	29.63	37.83	(30)		DFT	39.54	35.45	43.92	(19)
	$DFTW$	0.97	0.00	2.12	(20,8)		$DFTW$	3.49	1.32	6.35	(20,6)		$DFTW$	1.56	0.53	2.91	(20,9)
	ACF	0.43	0.00	1.59	(14)		ACF	0.41	0.00	1.32	(14)		ACF	0.35	0.00	1.06	(14)
	$POLY$	33.01	26.98	37.83	(20)		$POLY$	51.36	46.30	56.35	(1)		$POLY$	33.53	29.63	38.36	(20)
	SVD	55.96	50.79	59.52	(6)		SVD	56.12	52.12	60.05	(4)		SVD	75.00	70.63	78.04	(19)
	PCA	55.64	52.12	60.85	(7)		PCA	55.75	51.85	60.32	(6)		PCA	82.02	77.78	86.77	(20)
	WPE	4.75	2.91	7.41	(4)		WPE	2.54	1.32	4.76	(4)		WPE	5.00	2.65	7.67	(4)
	WPS	2.36	1.06	3.97	(4)		WPS	11.93	7.67	16.14	(2)		WPS	4.30	2.65	6.35	(4)
	$DWTS$	2.03	1.06	3.70	(3)		$DWTS$	15.93	11.11	20.11	(2)		$DWTS$	5.65	3.17	8.47	(3)
	DCT	67.16	65.08	69.31	(0.5)		DCT	84.39	84.39	84.39	(0.1)		DCT	20.78	17.20	24.34	(0.5)
	$DCT2$	40.29	35.98	44.18	(20)		$DCT2$	46.77	40.48	52.65	(20)		$DCT2$	52.11	46.56	57.94	(40)
	$DFT2$	1.27	0.26	2.91	(90)		$DFT2$	0.73	0.00	2.38	(100)		$DFT2$	0.88	0.00	2.38	(90)
	RAW	81.22	78.84	83.33	(1)		RAW	80.95	78.31	82.80	(1)		RAW	28.30	24.34	32.01	(1)
CH	SM	3.01	1.32	4.76	(1)	CR	SM	76.72	76.72	76.72	(1)						
	DWT	72.54	67.72	78.04	(1)		DWT	44.85	40.21	49.74	(6)						
	$CHEB$	42.46	37.57	47.88	(15)		$CHEB$	50.01	44.44	53.44	(19)						
	PAA	51.74	45.24	56.08	(5)		PAA	75.88	70.37	80.42	(20)						
	$ARMA$	1.24	0.00	3.17	(6,2)		$ARMA$	0.82	0.26	2.65	(2,5)						
	$ARIMA$	0.94	0.00	2.65	(4,6)		$ARIMA$	0.82	0.00	3.44	(2,6)						
	MSE	3.04	1.59	5.29	(14)		MSE	5.85	3.44	8.99	(14)						
	$FMSE$	32.24	26.46	36.24	(13)		$FMSE$	35.42	31.22	39.42	(15)						
	DFT	35.46	31.22	39.95	(19)		DFT	38.42	34.39	42.33	(19)						
	$DFTW$	2.97	1.32	5.82	(18,9)		$DFTW$	3.96	1.06	6.08	(20,8)						
	ACF	0.58	0.00	1.85	(15)		ACF	0.36	0.00	1.32	(15)						
	$POLY$	32.87	27.78	38.62	(20)		$POLY$	33.63	29.63	38.36	(20)						
	SVD	57.08	51.85	61.38	(4)		SVD	75.33	71.43	79.89	(20)						
	PCA	57.25	52.38	61.38	(6)		PCA	82.15	78.57	85.71	(19)						
	WPE	6.98	4.50	9.52	(4)		WPE	6.20	3.17	8.73	(4)						
	WPS	5.00	2.91	6.88	(4)		WPS	4.76	2.91	7.41	(4)						
	$DWTS$	2.96	1.32	6.61	(3)		$DWTS$	6.01	3.70	8.47	(3)						
	DCT	66.46	63.23	69.05	(0.5)		DCT	20.90	16.67	24.87	(0.5)						
	$DCT2$	38.75	32.80	42.59	(20)		$DCT2$	45.96	41.80	52.91	(40)						
	$DFT2$	5.12	2.91	7.94	(80)		$DFT2$	2.12	0.79	4.76	(100)						
	RAW	79.34	76.98	81.75	(1)		RAW	29.99	25.93	34.66	(1)						

Table A.4: Averaged classification error over 100 trials for SGD-based algorithms in the CWRU experiment

		SGDW1				SGDW2				SGDW3				SGDG1				SGDG2			
d	W	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node
ED	$db1$	0.33	0.00	1.59	(3,4)	7.30	5.03	9.26	(3,1)	0.62	0.00	1.32	(4,2)	2.25	0.79	4.23	(3,1)	8.43	5.82	12.17	(4,1)
	$db3$	0.33	0.00	1.59	(3,4)	5.50	3.70	7.41	(3,1)	0.63	0.00	1.85	(3,2)	1.91	0.79	3.17	(3,1)	11.00	8.47	14.29	(5,1)
	$db6$	0.37	0.00	1.06	(2,6)	4.70	2.91	6.88	(3,1)	0.75	0.00	1.85	(3,2)	1.89	0.79	3.17	(3,1)	13.19	10.32	16.40	(4,1)
	$bior1.3$	0.25	0.00	1.59	(3,4)	4.93	2.12	7.41	(3,1)	0.86	0.00	2.38	(2,2)	2.12	0.79	4.23	(3,1)	14.84	12.17	18.78	(0,2)
	$bior5.5$	0.39	0.00	1.59	(3,4)	5.71	3.70	8.47	(4,1)	0.80	0.00	1.85	(3,2)	2.59	1.06	3.97	(3,1)	15.05	11.90	19.58	(0,2)
	$coif1$	0.33	0.00	1.59	(3,4)	7.28	5.03	10.58	(2,1)	0.72	0.00	1.85	(3,2)	2.20	0.53	3.97	(3,1)	14.60	10.85	18.52	(1,1)
	$coif3$	0.32	0.00	1.59	(2,5)	6.79	4.23	9.26	(2,1)	0.48	0.00	1.59	(3,2)	1.85	0.26	3.44	(3,1)	13.30	10.32	16.67	(4,1)
	$sym2$	0.27	0.00	1.06	(2,5)	5.97	3.17	8.47	(3,1)	0.72	0.00	1.85	(1,3)	2.62	1.32	3.97	(3,1)	7.37	5.03	9.79	(2,1)
	$sym4$	0.19	0.00	0.79	(2,5)	3.48	1.85	6.61	(2,1)	0.68	0.00	1.85	(3,2)	2.19	0.53	3.97	(3,1)	14.45	10.85	18.52	(3,1)
SE	$db1$	0.31	0.00	1.32	(2,5)	7.13	4.23	8.99	(4,1)	6.22	3.97	10.05	(1,2)	15.18	11.11	18.25	(0,2)	8.14	5.03	11.11	(4,1)
	$db3$	0.27	0.00	1.06	(2,5)	5.25	2.38	8.47	(3,1)	7.06	3.97	10.58	(1,2)	15.10	11.90	18.78	(0,2)	10.06	8.20	13.76	(5,1)
	$db6$	0.24	0.00	1.32	(3,5)	5.07	2.12	7.14	(2,1)	7.08	4.50	9.26	(0,2)	15.32	11.38	19.84	(0,2)	11.59	9.26	14.02	(4,1)
	$bior1.3$	0.31	0.00	1.06	(2,5)	4.75	2.91	6.61	(3,1)	6.93	3.44	9.52	(1,2)	15.02	10.58	18.78	(0,2)	14.46	11.38	19.84	(0,2)
	$bior5.5$	0.24	0.00	1.32	(2,5)	6.27	3.97	8.99	(4,1)	7.21	4.76	9.79	(0,2)	15.16	11.38	19.58	(0,2)	14.49	10.85	18.25	(0,2)
	$coif1$	0.31	0.00	1.59	(2,5)	7.24	5.29	9.52	(2,1)	7.14	5.29	10.05	(0,2)	15.03	10.85	18.78	(0,2)	14.15	10.32	17.72	(2,1)
	$coif3$	0.21	0.00	1.59	(2,5)	6.16	4.50	8.47	(2,1)	7.17	5.03	9.52	(0,2)	15.52	12.17	19.05	(0,2)	13.65	10.58	17.46	(4,1)
	$sym2$	0.23	0.00	1.06	(2,5)	5.95	3.70	8.47	(3,1)	7.16	4.76	10.05	(0,2)	15.24	11.38	18.52	(0,2)	7.45	5.29	10.85	(2,1)
	$sym4$	0.24	0.00	1.85	(2,5)	3.79	2.12	5.82	(2,1)	6.51	4.23	8.73	(1,2)	15.04	10.85	17.99	(0,2)	14.50	11.64	18.78	(0,2)
CO	$db1$	0.59	0.00	2.38	(3,4)	8.25	6.08	11.11	(2,1)	0.58	0.00	1.32	(3,2)	2.19	0.79	3.70	(3,1)	8.70	6.08	11.11	(4,1)
	$db3$	0.49	0.00	1.59	(2,5)	7.07	4.76	10.05	(3,1)	0.66	0.00	1.59	(4,2)	1.94	0.53	3.44	(3,1)	11.40	8.99	14.02	(5,1)
	$db6$	0.35	0.00	1.59	(3,5)	6.25	3.17	8.73	(3,1)	0.56	0.00	1.59	(3,2)	1.92	0.53	3.44	(3,1)	13.60	10.85	17.46	(4,1)
	$bior1.3$	0.45	0.00	1.32	(3,5)	9.99	7.14	12.43	(5,1)	0.65	0.00	1.59	(2,2)	2.28	1.06	4.23	(3,1)	16.09	12.70	19.05	(4,1)
	$bior5.5$	0.62	0.00	2.38	(2,4)	10.77	7.67	13.23	(4,1)	0.69	0.00	1.59	(3,2)	2.37	0.79	3.70	(3,1)	16.38	13.76	19.05	(3,1)
	$coif1$	0.62	0.00	2.12	(3,4)	11.31	8.99	14.02	(2,1)	0.58	0.00	2.12	(3,2)	2.06	0.26	3.97	(3,1)	22.20	17.72	28.31	(4,1)
	$coif3$	0.33	0.00	1.32	(2,5)	8.88	6.08	11.64	(3,1)	0.49	0.00	1.59	(3,2)	1.74	0.53	3.44	(3,1)	19.22	15.87	22.22	(4,1)
	$sym2$	0.40	0.00	1.85	(2,5)	8.18	5.03	11.11	(4,1)	0.59	0.00	1.32	(3,2)	2.60	1.06	4.50	(3,1)	20.38	17.99	23.02	(4,1)
	$sym4$	0.47	0.00	1.32	(2,5)	5.42	3.97	7.41	(3,1)	0.67	0.00	1.59	(3,2)	2.07	0.26	3.97	(3,1)	22.98	20.63	25.93	(4,1)
CH	$db1$	0.51	0.00	1.85	(3,4)	7.61	5.29	10.32	(3,1)	1.52	0.00	2.91	(3,2)	3.06	1.32	6.61	(0,1)	8.70	5.82	11.64	(4,1)
	$db3$	0.50	0.00	1.85	(3,4)	6.23	4.23	9.26	(3,1)	1.23	0.26	2.38	(2,2)	2.83	1.06	4.50	(3,1)	11.99	8.99	14.81	(4,1)
	$db6$	0.54	0.00	1.59	(2,4)	5.78	3.44	7.94	(3,1)	0.85	0.00	2.12	(2,2)	2.99	1.32	5.29	(0,1)	13.74	10.32	18.25	(1,1)
	$bior1.3$	0.46	0.00	1.85	(3,4)	5.44	3.70	8.47	(3,1)	1.34	0.26	2.38	(2,2)	2.89	1.32	4.50	(0,1)	14.87	11.38	17.99	(0,2)
	$bior5.5$	0.44	0.00	1.32	(2,4)	6.68	3.97	10.05	(3,1)	1.27	0.26	3.44	(1,2)	3.08	1.06	5.03	(0,1)	14.66	11.38	17.99	(0,2)
	$coif1$	0.51	0.00	1.59	(2,4)	7.15	5.03	10.05	(2,1)	1.52	0.26	2.91	(2,2)	2.96	1.59	5.29	(0,1)	14.45	11.64	17.46	(0,2)
	$coif3$	0.49	0.00	1.59	(2,5)	7.24	3.70	8.99	(2,1)	1.03	0.00	2.12	(3,2)	3.01	1.06	5.56	(0,1)	13.75	10.05	16.93	(4,1)
	$sym2$	0.52	0.00	2.12	(3,4)	6.68	3.97	9.52	(3,1)	0.98	0.26	2.38	(1,2)	3.11	1.32	4.76	(0,1)	7.31	5.56	9.79	(2,1)
	$sym4$	0.54	0.00	1.85	(2,5)	3.76	2.12	5.56	(2,1)	1.44	0.53	2.65	(1,2)	3.07	1.06	5.29	(0,1)	14.59	12.17	17.72	(0,2)
CR	$db1$	0.54	0.00	2.38	(3,4)	8.34	4.76	12.17	(2,1)	0.57	0.00	1.32	(5,2)	2.35	0.53	4.50	(3,1)	8.46	5.29	12.17	(4,1)
	$db3$	0.47	0.00	1.85	(2,5)	6.94	5.29	9.52	(3,1)	0.69	0.00	1.85	(3,2)	1.95	0.79	3.97	(3,1)	11.42	8.73	14.55	(5,1)
	$db6$	0.46	0.00	2.12	(3,5)	6.36	4.50	8.99	(3,1)	0.57	0.00	1.85	(3,2)	2.08	1.06	3.70	(3,1)	13.68	10.32	17.46	(4,1)
	$bior1.3$	0.44	0.00	1.59	(3,4)	9.99	7.14	13.49	(5,1)	0.77	0.00	2.38	(2,2)	2.39	1.06	3.97	(3,1)	16.25	12.96	21.69	(4,1)
	$bior5.5$	0.56	0.00	2.38	(2,4)	10.60	8.47	12.96	(4,1)	0.73	0.00	2.38	(3,2)	2.34	1.06	4.23	(3,1)	16.43	13.49	19.58	(3,1)
	$coif1$	0.63	0.00	2.12	(3,4)	11.22	8.47	14.55	(2,1)	0.66	0.00	2.12	(3,2)	2.23	0.79	3.97	(3,1)	24.17	19.84	28.57	(4,1)
	$coif3$	0.38	0.00	1.59	(2,5)	8.98	6.88	12.70	(3,1)	0.49	0.00	1.32	(3,2)	2.06	0.79	4.23	(3,1)	18.49	15.08	21.43	(2,2)
	$sym2$	0.40	0.00	1.06	(2,5)	8.10	5.82	11.38	(4,1)	0.63	0.00	1.59	(3,2)	2.37	1.06	4.50	(3,1)	20.68	17.99	24.34	(4,1)
	$sym4$	0.49	0.00	1.59	(2,5)	5.28	3.44	7.41	(3,1)	0.67	0.00	1.85	(3,2)	2.40	0.79	4.50	(3,1)	23.19	20.63	25.40	(4,1)

Table A.5: Classification error over 100 trials for benchmark representations in ECG experiment

d	Method	Avg	Min	Max	Node	*d*	Method	Avg	Min	Max	Node	*d*	Method	Avg	Min	Max	Node
ED	*SM*	19.69	15.94	25.31	(1)	SE	*SM*	20.03	16.25	24.06	(1)	CO	*SM*	58.78	54.37	62.81	(1)
	DWT	78.59	74.06	82.19	(2)		*DWT*	66.84	61.56	73.12	(2)		*DWT*	82.16	78.75	86.88	(2)
	CHEB	60.87	56.56	66.25	(3)		*CHEB*	67.81	61.56	75.00	(2)		*CHEB*	80.81	75.94	85.62	(20)
	PAA	63.78	58.44	70.94	(8)		*PAA*	64.16	59.69	70.00	(8)		*PAA*	78.93	73.75	84.06	(16)
	ARMA	33.06	28.12	38.12	(3,1)		*ARMA*	93.75	93.75	93.75	(1,1)		*ARMA*	29.80	25.00	36.56	(4,1)
	ARIMA	39.74	33.44	47.50	(2,1)		*ARIMA*	93.75	93.75	93.75	(1,1)		*ARIMA*	39.28	33.75	45.62	(2,1)
	MSE	35.34	30.63	40.31	(15)		*MSE*	34.88	30.31	40.00	(15)		*MSE*	55.91	50.62	60.00	(15)
	FMSE	65.78	60.31	72.81	(15)		*FMSE*	63.22	57.19	68.44	(15)		*FMSE*	74.17	69.06	78.44	(15)
	DFT	61.41	56.56	65.62	(28)		*DFT*	61.66	55.62	68.44	(27)		*DFT*	72.70	67.81	77.81	(27)
	DFTW	32.77	28.75	37.50	(20,7)		*DFTW*	26.73	22.81	31.56	(20,7)		*DFTW*	48.56	41.88	54.37	(20,9)
	ACF	7.58	4.69	10.31	(15)		*ACF*	7.63	5.31	11.25	(15)		*ACF*	7.64	5.31	11.88	(15)
	POLY	58.93	53.44	65.00	(14)		*POLY*	55.51	50.00	60.94	(14)		*POLY*	63.60	57.81	69.06	(10)
	SVD	70.68	67.19	75.62	(1)		*SVD*	67.26	61.88	71.88	(2)		*SVD*	84.27	79.06	87.19	(14)
	PCA	84.07	80.00	89.38	(4)		*PCA*	83.86	79.06	87.19	(4)		*PCA*	91.31	87.81	94.38	(12)
	WPE	33.21	29.06	37.50	(5)		*WPE*	25.13	20.94	30.00	(6)		*WPE*	27.61	22.81	31.56	(6)
	WPS	18.17	13.75	22.19	(3)		*WPS*	21.20	15.94	25.62	(1)		*WPS*	35.04	30.63	41.88	(3)
	DWTS	21.90	17.50	27.50	(3)		*DWTS*	21.63	17.19	29.69	(1)		*DWTS*	35.86	30.63	42.50	(4)
	DCT	81.03	76.25	88.12	(0.5)		*DCT*	93.75	93.75	93.75	(0.1)		*DCT*	90.30	87.81	92.50	(0.5)
	*DCT*2	62.65	57.19	69.06	(10)		*DCT*2	71.68	65.31	76.56	(30)		*DCT*2	71.36	64.69	75.94	(50)
	*DFT*2	20.87	15.94	27.81	(30)		*DFT*2	27.05	22.19	31.87	(60)		*DFT*2	32.23	26.56	38.44	(100)
	RAW	92.00	90.00	93.75	(1)		*RAW*	91.49	89.06	93.75	(1)		*RAW*	92.48	90.94	93.75	(1)
CH	*SM*	19.85	15.31	23.12	(1)	CR	*SM*	93.78	92.81	94.38	(1)						
	DWT	84.95	80.00	90.00	(2)		*DWT*	81.96	77.81	85.62	(2)						
	CHEB	60.38	55.62	65.00	(3)		*CHEB*	80.63	77.19	84.69	(20)						
	PAA	65.04	60.62	70.94	(3)		*PAA*	83.52	79.69	88.75	(16)						
	ARMA	32.36	26.88	38.12	(3,1)		*ARMA*	30.86	25.94	36.56	(4,1)						
	ARIMA	40.97	36.56	46.88	(2,1)		*ARIMA*	40.71	36.25	47.19	(2,1)						
	MSE	40.58	35.00	46.56	(15)		*MSE*	69.33	64.38	73.75	(15)						
	FMSE	71.19	66.56	75.31	(15)		*FMSE*	78.42	73.44	83.12	(15)						
	DFT	65.71	59.69	70.62	(28)		*DFT*	73.30	69.69	78.75	(27)						
	DFTW	41.31	36.25	47.19	(17,6)		*DFTW*	63.03	57.81	68.12	(18,7)						
	ACF	7.65	5.00	10.94	(14)		*ACF*	6.68	3.75	10.00	(14)						
	POLY	58.45	54.06	65.31	(14)		*POLY*	63.44	58.75	69.38	(10)						
	SVD	70.91	65.62	75.62	(1)		*SVD*	85.27	80.94	89.38	(14)						
	PCA	83.61	78.44	88.44	(4)		*PCA*	91.62	88.12	94.06	(11)						
	WPE	36.19	31.87	41.88	(5)		*WPE*	27.65	23.44	33.44	(6)						
	WPS	22.06	19.06	25.94	(1)		*WPS*	34.91	30.63	41.25	(3)						
	DWTS	22.15	17.50	25.62	(1)		*DWTS*	35.62	28.44	40.00	(4)						
	DCT	81.77	78.44	86.25	(0.5)		*DCT*	90.27	86.88	92.50	(0.5)						
	*DCT*2	62.98	58.44	68.75	(10)		*DCT*2	70.77	65.00	75.62	(50)						
	*DFT*2	24.73	19.69	29.06	(30)		*DFT*2	41.95	35.00	48.44	(30)						
	RAW	89.33	87.19	91.88	(1)		*RAW*	85.16	80.94	89.69	(1)						

Table A.6: Averaged classification error over 100 trials for SGD-based algorithms in the ECG biometrics experiment

		SGDW1				**SGDW2**				**SGDW3**				**SGDG1**				**SGDG2**			
d	W	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node	Avg	Min	Max	Node
ED	*db*1	1.81	0.31	3.12	(1,4)	7.21	4.06	11.25	(2,2)	0.33	0.00	0.94	(1,5)	11.96	9.38	15.94	(0,4)	9.66	6.25	13.75	(3,2)
	*db*3	1.42	0.31	3.12	(2,4)	6.51	3.44	10.00	(2,2)	0.46	0.00	1.56	(1,5)	12.22	8.12	17.50	(0,4)	9.23	5.62	12.81	(3,2)
	*db*6	1.44	0.31	3.44	(2,4)	9.53	5.62	13.44	(2,2)	0.77	0.00	1.88	(1,5)	12.41	8.75	16.25	(0,4)	13.35	9.69	16.56	(3,2)
	*bior*1.3	1.66	0.31	3.12	(2,4)	7.39	3.75	10.31	(2,2)	0.34	0.00	0.94	(1,5)	12.22	9.38	16.88	(0,4)	9.74	6.56	13.44	(3,2)
	*bior*5.5	1.31	0.31	3.12	(2,4)	5.03	2.81	7.19	(3,2)	0.96	0.00	1.88	(0,6)	12.16	8.44	15.94	(0,4)	15.68	11.56	19.38	(0,2)
	*coif*1	1.33	0.00	4.06	(1,5)	6.91	4.69	9.69	(2,2)	0.44	0.00	1.25	(1,5)	12.07	7.81	16.25	(0,4)	12.31	9.69	17.19	(1,2)
	*coif*3	1.64	0.31	3.44	(1,6)	9.31	5.94	13.44	(2,2)	0.55	0.00	1.88	(1,5)	11.99	8.12	16.25	(0,4)	15.58	11.56	20.00	(0,2)
	*sym*2	1.02	0.00	3.44	(2,4)	6.27	3.75	10.31	(3,2)	0.36	0.00	1.88	(1,5)	11.91	7.81	16.88	(0,4)	12.25	9.38	15.00	(3,2)
	*sym*4	1.38	0.00	3.75	(2,4)	8.24	5.00	11.25	(1,1)	0.58	0.00	1.25	(1,5)	12.24	8.75	15.94	(0,4)	15.73	12.81	19.06	(0,2)
SE	*db*1	0.49	0.00	1.88	(1,5)	13.52	10.62	17.50	(0,2)	0.29	0.00	1.56	(0,5)	5.52	3.12	9.38	(0,2)	7.83	5.31	11.25	(3,2)
	*db*3	1.72	0.62	3.12	(0,4)	13.34	9.06	17.81	(0,2)	0.30	0.00	1.56	(0,6)	5.37	2.19	8.44	(0,2)	6.71	3.75	11.25	(3,2)
	*db*6	1.68	0.62	3.12	(0,4)	13.23	9.38	17.19	(0,2)	0.31	0.00	1.25	(0,6)	5.51	2.81	10.31	(0,2)	12.89	9.38	16.88	(0,2)
	*bior*1.3	0.54	0.00	1.56	(1,5)	11.48	6.88	93.75	(1,1)	0.33	0.00	1.88	(0,6)	5.36	2.19	9.38	(0,2)	8.85	5.31	13.12	(3,2)
	*bior*5.5	1.65	0.31	3.12	(0,4)	13.61	10.31	16.88	(0,2)	0.29	0.00	1.88	(0,5)	5.59	3.12	9.38	(0,2)	12.78	9.06	15.94	(0,2)
	*coif*1	1.78	0.31	4.06	(0,4)	13.41	9.69	17.19	(0,2)	0.29	0.00	2.19	(0,6)	5.33	2.81	8.44	(0,2)	7.50	4.69	10.94	(1,2)
	*coif*3	1.70	0.31	3.44	(0,4)	13.22	9.38	18.44	(0,2)	0.31	0.00	1.56	(0,5)	5.33	2.81	9.06	(0,2)	12.97	8.75	17.81	(0,2)
	*sym*2	1.75	0.62	3.75	(0,4)	13.53	9.06	18.12	(0,2)	0.32	0.00	1.25	(0,5)	5.43	2.19	9.69	(0,2)	9.90	6.56	13.44	(2,2)
	*sym*4	1.86	0.62	3.75	(0,4)	13.30	8.75	17.19	(0,2)	0.31	0.00	1.88	(0,6)	5.39	2.81	8.44	(0,2)	12.84	9.69	16.56	(0,2)
CO	*db*1	1.56	0.31	3.44	(1,4)	7.33	4.06	11.56	(2,2)	0.27	0.00	0.94	(1,5)	12.14	8.75	15.62	(1,2)	9.28	5.31	12.50	(3,2)
	*db*3	1.70	0.62	3.75	(3,3)	6.02	3.44	10.00	(1,1)	0.62	0.00	1.56	(1,5)	12.73	9.06	16.88	(0,4)	9.21	5.62	12.19	(3,2)
	*db*6	1.87	0.00	3.44	(3,3)	11.93	7.81	15.94	(3,2)	0.95	0.00	1.88	(1,5)	12.94	10.00	15.94	(0,4)	13.18	6.88	17.50	(3,2)
	*bior*1.3	1.45	0.31	3.12	(2,4)	8.82	5.00	13.12	(1,1)	0.38	0.00	1.56	(1,5)	12.39	8.75	16.56	(1,2)	9.86	5.94	13.44	(3,2)
	*bior*5.5	1.72	0.00	4.06	(2,4)	6.56	4.06	10.00	(3,2)	1.02	0.00	2.19	(0,5)	12.81	8.12	16.88	(0,4)	15.79	11.56	20.62	(1,2)
	*coif*1	1.66	0.31	4.38	(1,5)	5.91	3.12	10.62	(1,1)	0.55	0.00	1.88	(1,5)	12.91	7.19	16.25	(0,4)	12.94	9.38	16.88	(1,2)
	*coif*3	2.15	0.31	5.00	(2,4)	10.95	7.81	14.37	(1,1)	0.72	0.00	1.88	(1,5)	12.77	8.75	17.50	(0,4)	16.05	11.56	21.25	(1,2)
	*sym*2	1.24	0.00	2.81	(2,4)	6.93	3.75	10.31	(3,2)	0.51	0.00	1.88	(1,5)	12.45	8.75	16.88	(0,4)	12.60	9.69	15.94	(3,2)
	*sym*4	2.07	0.31	5.00	(1,5)	8.58	5.62	12.19	(2,2)	0.80	0.00	2.81	(1,5)	12.69	8.44	16.88	(0,4)	16.07	12.19	20.00	(0,2)
CH	*db*1	3.51	1.56	6.25	(1,4)	9.28	6.56	12.19	(2,2)	1.32	0.31	3.12	(1,3)	13.70	10.00	17.19	(0,4)	12.46	9.06	15.62	(3,2)
	*db*3	3.07	0.94	5.00	(2,3)	9.25	5.94	13.44	(2,2)	2.87	1.25	5.94	(1,2)	13.80	9.38	18.44	(0,4)	12.23	8.12	18.12	(3,2)
	*db*6	4.31	1.88	6.56	(1,4)	11.95	8.75	16.25	(2,2)	2.88	0.94	7.50	(0,4)	13.96	10.31	20.31	(0,4)	15.73	11.56	20.00	(3,2)
	*bior*1.3	3.58	1.56	6.56	(1,4)	9.62	6.25	12.50	(2,2)	1.29	0.00	3.44	(1,3)	14.03	10.94	17.50	(0,4)	12.17	8.44	18.44	(3,2)
	*bior*5.5	3.94	2.19	5.94	(2,3)	7.24	4.69	11.88	(1,2)	2.92	0.62	5.62	(0,4)	14.07	10.31	18.12	(0,4)	17.26	12.50	21.56	(0,2)
	*coif*1	3.94	2.19	6.25	(2,3)	9.13	5.31	12.50	(1,1)	2.30	1.25	4.69	(1,2)	14.15	10.31	17.81	(0,4)	14.05	10.00	18.75	(1,2)
	*coif*3	4.24	2.19	6.56	(1,4)	12.57	9.38	16.56	(2,2)	2.76	0.31	5.94	(0,4)	13.96	9.06	18.75	(0,4)	16.85	11.88	21.56	(0,2)
	*sym*2	3.46	1.56	5.94	(2,3)	9.30	5.00	13.44	(1,1)	2.16	0.94	4.69	(1,2)	14.17	10.31	19.06	(0,4)	14.85	10.94	18.75	(3,2)
	*sym*4	3.74	1.88	6.25	(2,3)	10.13	5.00	14.69	(1,1)	2.96	0.94	5.94	(0,4)	13.80	8.44	18.12	(0,4)	16.82	12.81	20.62	(0,2)
CR	*db*1	1.61	0.31	3.75	(1,4)	7.12	3.75	10.62	(2,2)	0.31	0.00	0.94	(1,5)	12.73	8.75	16.56	(0,4)	9.48	6.56	13.44	(3,2)
	*db*3	1.66	0.31	3.44	(3,3)	5.99	2.81	8.44	(1,1)	0.64	0.00	1.88	(1,5)	12.88	9.38	17.19	(0,4)	8.83	5.00	11.88	(3,2)
	*db*6	1.77	0.31	4.38	(3,3)	12.07	8.75	16.25	(3,2)	1.01	0.00	2.81	(1,5)	13.07	9.38	16.25	(0,4)	12.94	9.06	17.19	(3,2)
	*bior*1.3	1.29	0.31	3.12	(2,4)	8.62	5.94	12.50	(2,2)	0.32	0.00	0.94	(1,5)	12.16	8.75	16.88	(1,3)	10.42	7.19	15.62	(3,2)
	*bior*5.5	1.70	0.31	3.12	(2,4)	6.38	3.44	9.38	(3,2)	1.13	0.00	2.19	(0,6)	12.84	9.06	15.94	(0,4)	15.70	11.56	19.69	(1,2)
	*coif*1	1.57	0.31	3.44	(1,5)	6.15	3.44	10.00	(1,1)	0.62	0.00	1.56	(1,5)	13.13	9.69	17.50	(0,4)	13.29	8.75	16.88	(3,2)
	*coif*3	2.23	0.94	5.00	(2,4)	10.79	7.50	15.00	(1,1)	0.79	0.00	2.50	(1,5)	12.91	9.06	17.19	(0,4)	16.48	10.31	21.56	(1,2)
	*sym*2	1.26	0.00	2.50	(2,4)	6.74	4.69	9.69	(3,2)	0.55	0.00	1.56	(1,5)	12.65	8.75	16.56	(0,4)	12.53	9.69	17.50	(3,2)
	*sym*4	2.10	0.62	4.06	(2,4)	8.82	5.94	13.75	(2,2)	0.68	0.00	1.88	(1,5)	13.06	9.38	18.12	(0,4)	15.93	11.88	20.00	(2,2)

Table A.7: Classification error over 100 trials for UCR Experiment using ED, SE and CO as distance measures.

	ED		**SE**		**CO**	
Data set	Avg	Node	Avg	Node	Avg	Node
50w	30.60	(PAA,18)	30.39	(PAA,18)	31.38	(PAA,18)
Adc	22.55	($SGDW2$,$db3$,1,4)	23.28	($SGDW2$,$db6$,0,4)	22.58	($SGDW2$,$db3$,1,4)
Beef	22.43	(PCA,10)	22.63	(SVD,16)	18.50	(PCA,10)
CBF	0.00	($CHEB$,20)	0.02	(PAA,16)	0.01	($CHEB$,17)
ChlC	1.93	(SVD,19)	1.93	(SVD,19)	1.97	(PCA,19)
Cinc	0.14	(SVD,20)	0.12	(SVD,20)	0.13	(SVD,18)
Coff	0.96	(ACF,15)	1.33	(ACF,14)	0.22	(ACF,12)
CriX	31.82	($SGDW3$,$db1$,2,3)	29.62	($SGDW3$,$db3$,0,3)	29.82	($SGDW3$,$db1$,2,3)
CriY	32.25	($SGDW3$,$db3$,6,4)	32.14	($SGDW3$,$db3$,0,3)	29.91	($SGDW3$,$db1$,2,3)
CriZ	31.99	($SGDW3$,$db1$,1,4)	28.79	($SGDW3$,$db6$,0,3)	29.40	($SGDW3$,$db3$,1,3)
DiSR	0.07	($SGDW1$,$db6$,2,2)	0.16	(DWT,1)	0.14	($SGDW1$,$db6$,2,2)
ECG2	0.00	(SM,1)	0.00	(SM,1)	0.00	($SGDW4$,$db1$,0,1)
ECG5	0.01	($DFT2$,40)	0.00	($DFT2$,10)	0.00	($DFT2$,40)
Fish	16.78	(PCA,16)	17.49	(SVD,16)	16.55	(SVD,16)
FacA	6.29	($DCT2$,30)	5.80	(DWT,1)	6.28	($CHEB$,19)
Fac4	9.29	(PCA,15)	4.84	(PAA,18)	8.73	(PAA,18)
FacU	6.33	($DCT2$,30)	5.44	(DWT,2)	6.27	($CHEB$,20)
GunP	3.31	($DFT2$,60)	4.02	(DFT,13)	3.25	($DFT2$,80)
Hapt	48.99	($SGDW3$,$db1$,4,3)	49.32	($SGDW5$,$db3$,5,2)	48.37	($SGDW3$,$db1$,4,4)
InLS	15.57	($ARIMA$,6)	31.47	(WPE,5)	14.59	($ARIMA$,6)
Ltg2	17.59	($SGDW3$,$db6$,5,2)	19.85	($SGDW5$,$db6$,6,1)	17.64	($SGDW4$,$db6$,5,1)
Ltg7	28.83	($DCT2$,10)	29.84	(PAA,17)	32.54	(DWT,2)
Mall	1.46	($SGDW3$,$db3$,3,4)	2.14	(SVD,5)	1.41	($SGDW3$,$db6$,4,3)
MedI	21.16	($SGDW2$,$db3$,6,2)	21.88	($SGDW3$,$db3$,6,2)	21.42	($SGDW2$,$db3$,6,2)
MotS	5.80	(PAA,16)	6.08	(PAA,16)	4.87	(DWT,1)
OsuL	25.34	($SGDW3$,$db3$,2,3)	21.60	($SGDW3$,$db3$,6,2)	24.60	($SGDW3$,$db3$,2,3)
OliO	9.38	($SGDW1$,$db1$,6,1)	11.26	($SGDW5$,$db1$,5,2)	9.93	($SGDW1$,$db1$,6,1)
SnS2	2.21	(PAA,20)	2.13	($DCT2$,30)	2.30	($DCT2$,50)
SnS	0.95	(PAA,14)	1.44	(PAA,14)	1.62	(PCA,14)
StLC	2.92	($SGDW3$,$db1$,5,3)	3.44	($SGDW3$,$db1$,2,3)	2.91	($SGDW3$,$db1$,5,3)
SweL	12.09	($SGDW2$,$db6$,4,2)	12.64	(DFT,27)	12.17	($SGDW2$,$db6$,4,2)
Symb	1.95	($SGDW3$,$db1$,4,2)	2.08	($SGDW3$,$db1$,3,2)	2.30	($SGDW3$,$db1$,3,2)
Trce	0.00	($POLY$,3)	0.00	($POLY$,3)	0.00	($POLY$,3)
2ECG	0.16	(SVD,15)	0.12	(PCA,15)	0.10	(SVD,15)
TWoP	1.19	($DCT2$,20)	2.03	($CHEB$,19)	1.22	(PAA,20)
WorS	30.30	(PAA,18)	30.16	(PAA,19)	30.67	(DWT,1)
SynC	0.84	($DCT2$,20)	0.75	(DWT,1)	1.54	($SGDW3$,$db1$,6,3)
uWGX	23.54	(PAA,15)	23.65	(PAA,15)	23.76	($DCT2$,20)
uWGY	29.25	($DCT2$,20)	29.47	(PAA,20)	29.39	($DCT2$,20)
uWGZ	29.93	($DCT2$,10)	29.85	(PAA,13)	30.36	(DWT,1)
Wafr	0.12	($DFT2$,90)	0.10	($DFT2$,30)	0.13	($DFT2$,100)
Yoga	7.32	(DWT,4)	6.78	(DWT,2)	7.25	(DWT,4)

Table A.8: Classification error over 100 trials for UCR Experiment using CH and CR as distance measures.

	CH		**CR**	
Data set	Avg	Node	Avg	Node
50w	32.25	($CHEB$,16)	31.36	(PAA,18)
Adc	24.38	($SGDW$2,db3,1,4)	22.45	($SGDW$2,db3,1,4)
Beef	29.47	(PCA,10)	20.33	(PCA,16)
CBF	0.01	($CHEB$,15)	0.01	($CHEB$,17)
ChlC	2.09	(PCA,19)	2.01	(SVD,19)
Cinc	0.19	(PCA,18)	0.11	(PCA,17)
Coff	1.04	(ACF,12)	0.00	(ACF,8)
CriX	33.19	($SGDW$3,db3,1,4)	29.62	($SGDW$3,db1,2,3)
CriY	33.99	($SGDW$3,db1,1,3)	29.67	($SGDW$3,db1,1,3)
CriZ	33.57	($SGDW$3,db3,1,4)	29.58	($SGDW$3,db3,1,3)
DiSR	0.14	($SGDW$1,db6,2,2)	0.14	($SGDW$1,db6,2,2)
ECG2	0.00	(SM,1)	0.00	($SGDW$4,db1,0,1)
ECG5	0.07	(DFT2,100)	0.01	(DFT2,30)
Fish	21.32	(SVD,16)	17.29	(SVD,20)
FacA	9.86	($CHEB$,19)	6.35	($CHEB$,19)
Fac4	11.95	(DCT,0.25)	8.96	(PAA,18)
FacU	9.85	($CHEB$,19)	6.59	(DCT2,40)
GunP	5.79	(DFT2,90)	3.40	(DFT2,30)
Hapt	50.22	($SGDW$3,db1,4,3)	49.09	($SGDW$3,db1,4,4)
InLS	16.11	($ARIMA$,6)	15.36	($ARIMA$,6)
Ltg2	16.61	($SGDW$3,db6,5,1)	17.76	($SGDW$3,db6,5,2)
Ltg7	31.30	(DCT2,40)	32.19	(DWT,2)
Mall	1.96	($SGDW$3,db3,3,3)	1.47	($SGDW$3,db6,4,3)
MedI	23.94	($SGDW$2,db6,6,2)	21.46	($SGDW$2,db3,6,2)
MotS	6.58	(PAA,11)	4.83	(DWT,1)
OsuL	31.25	($SGDW$3,db3,3,2)	24.47	($SGDW$3,db3,2,2)
OliO	10.43	($SGDW$1,db1,6,4)	9.90	($SGDW$1,db1,5,2)
SnS2	3.73	(PAA,14)	2.18	(RAW,1)
SnS	1.92	(PCA,6)	1.62	(DCT2,40)
StLC	3.31	($SGDW$3,db1,4,2)	2.89	($SGDW$3,db1,5,3)
SweL	14.52	($SGDW$2,db6,5,2)	12.09	($SGDW$2,db6,4,2)
Symb	2.90	($SGDW$2,db6,6,1)	2.32	($SGDW$3,db1,3,2)
Trce	0.01	($POLY$,3)	0.00	(WPE,4)
2ECG	0.26	(SVD,15)	0.07	(PCA,15)
TWoP	2.36	(DCT2,30)	1.24	(DCT2,20)
WorS	31.63	($CHEB$,18)	30.75	(DWT,1)
SynC	1.22	($CHEB$,15)	1.69	($SGDW$3,db1,6,4)
uWGX	24.83	($CHEB$,14)	23.67	(DCT2,20)
uWGY	31.24	(PAA,14)	29.27	(DCT2,20)
uWGZ	31.23	(DCT2,10)	30.26	(DWT,1)
Wafr	0.16	($CHEB$,20)	0.10	(DFT2,80)
Yoga	9.31	(PAA,18)	7.35	(DWT,4)

Bibliography

[1] L. Zhang, G. Xiong, H. Liu, H. Zou, and W. Guo, "Applying improved multi-scale entropy and support vector machines for bearing health condition identification," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 224, no. 6, pp. 1315–1325, 2010.

[2] T. Liao, "Clustering of time series data–a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857 – 1874, 2005.

[3] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 2, pp. 307 –320, 2011.

[4] P. Foggia, C. Sansone, F. Tortorella, and M. Vento, "Combining statistical and structural approaches for handwritten character description," *Image and Vision Computing*, vol. 17, no. 9, pp. 701–711, 1999.

[5] R. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," Ph.D. dissertation, Carnegie Mellon University, 2001.

[6] H. Kriegel, K. Borgwardt, P. Kroger, A. Pryakhin, M. Schubert, and A. Zimek, "Future trends in data mining," *Data Mining and Knowledge Discovery*, vol. 15, pp. 87–97, 2007.

[7] T. Jebara, *Machine Learning: Discriminative and Generative.* Kluwer Academic Publishers, 2004.

[8] K. S. Fu, "A step towards unification of syntactic and statistical pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 3, pp. 398–404, 1986.

[9] C. Aggarwal, "On change diagnosis in evolving data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 587 – 600, 2005.

[10] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, no. 0, pp. 72 – 83, 2013.

[11] Y. Kawahara and M. Sugiyama, "Change-point detection in time-series data by direct density-ratio estimation," in *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)*, 2009, pp. 389–400.

[12] F. Gustafsson and F. Gustafsson, *Adaptive filtering and change detection.* Wiley London, 2000, vol. 1.

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

[14] R. Duin and E. Pekalska, "The science of pattern recognition. achievements and perspectives," in *Challenges for Computational Intelligence*, ser. Studies in Computational Intelligence, W. Duch and J. Mandziuk, Eds. Springer Berlin / Heidelberg, 2007, vol. 63, pp. 221–259.

[15] A. K. Jain, R. P. W. Duin, and M. Jianchang, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[16] L. Kanal, "Patterns in pattern recognition: 1968-1974," *IEEE Transactions on Information Theory*, vol. 20, no. 6, pp. 697–722, 1974.

[17] K. Fu and P. Swain, "On syntactic pattern recognition," in *Software Engineering*, J. Tou, Ed. New York: Academic Press, 1971, vol. 2.

[18] W. Tsai, "Combining statistical and structural methods," in *Syntactic and Structural Pattern Recognition: Theory and Applications*, ser. World Scientific Series in Computer Science, H. Bunke and A. Sanfeliu, Eds. World Scientific, 1990, pp. 349–366.

[19] V. Vapnik, *The nature of statistical learning theory.* Springer (New York), 1995.

[20] K. Fu, *Syntactic pattern recognition and applications.* Prentice Hall, 1982.

[21] W. Tsai and K. Fu, “A pattern deformational model and bayes error-correcting recognition system,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 12, pp. 745–756, 1979.

[22] H. C. Lee and F. King-Sun, “A stochastic syntax analysis procedure and its application to pattern classification,” *IEEE Transactions on Computers*, vol. C-21, no. 7, pp. 660–666, 1972.

[23] W. Tsai and K. Fu, “Attributed grammar-a tool for combining syntactic and statistical approaches to pattern recognition,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 10, no. 12, pp. 873–885, 1980.

[24] H. Nishida, “Shape recognition by integrating structural descriptions and geometrical/statistical transforms,” *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 248–262, 1996.

[25] L. Heutte, T. Paquet, J. V. Moreau, Y. Lecourtier, and C. Olivier, “A structural/statistical feature based vector for handwritten character recognition,” *Pattern Recognition Letters*, vol. 19, no. 7, pp. 629–641, 1998.

[26] H. Baird, “Feature identification for hybrid structural/statistical pattern classification,” *Computer Vision, Graphics, and Image Processing*, vol. 42, no. 3, pp. 318–333, 1988.

[27] Q. Xiao and H. Raafat, “Combining statistical and structural information for fingerprint image processing classification and identification,” in *Pattern recognition: architectures, algorithms and applications*, ser. World Scientific Series in Computer Science, R. Plamondon and H. Cheng, Eds. World Scientific, 1991, pp. 335–354.

[28] M. Halle and K. Stevens, “Speech recognition: A model and a program for research,” *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 155–159, 1962.

[29] V. Nair, J. Susskind, and G. Hinton, “Analysis-by-synthesis by learning to invert generative black boxes,” in *Artificial Neural Networks-ICANN 2008*. Springer, 2008, pp. 971–981.

[30] D. Hand, H. Mannila, and P. Smyth, *Principles of data mining.* MIT press, 2001.

[31] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," *Neural Information Processing Systems*, vol. 2, pp. 841–848, 2002.

[32] B. Ng, "Survey of anomaly detection methods," Lawrence Livermore National Laboratory (LLNL), Livermore, CA, Tech. Rep. UCRL-TR-225264, 2006.

[33] J. Milgram, R. Sabourin, and M. Cheriet, "Combining model-based and discriminative approaches in a modular two-stage classification system: Application to isolated handwritten digit recognition," *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, no. 2, p. 115, 2005.

[34] K. T. Abou-Moustafa, M. Cheriet, and C. Y. Suen, "Classification of time-series data using a generative/discriminative hybrid," in *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, 2004, pp. 51–56.

[35] L. Le Quan and S. Bengio, "Hybrid generative-discriminative models for speech and speaker recognition," IDIAP Research Institute, Tech. Rep. Idiap-RR-06-2002, 2002.

[36] T. Zhuowen, "Learning generative models via discriminative approaches," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.

[37] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, 2006, pp. 87–94.

[38] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[39] A. Holub, M. Welling, and P. Perona, "Hybrid generative-discriminative visual categorization," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 239–258, 2008.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[41] A. Bosch, A. Zisserman, and X. Muoz, “Scene classification using a hybrid generative/discriminative approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712–727, 2008.

[42] E. Awad and H. Ghaziri, *Knowledge Management.* Prentice-Hall, Upper Saddle River, New Jersey, 2004.

[43] K. Laudon and J. Laudon, *Management Information Systems: Managing the Digital Firm*, 7th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[44] N. Kakabadse, A. Kakabadse, and A. Kouzmin, “Reviewing the knowledge management literature: towards a taxonomy,” *Journal of Knowledge Management*, vol. 7, no. 4, pp. 75–91, 2003.

[45] G. Curtis and D. Cobham, *Business information systems: analysis, design and practice.* Pearson Education, 2008.

[46] J. Rowley, “The wisdom hierarchy: representations of the DIKW hierarchy,” *Journal of Information Science*, vol. 33, no. 2, pp. 163–180, 2007.

[47] T. Groff and T. Jones, *Introduction to knowledge management: KM in business.* Butterworth Heinemann, 2003.

[48] C. Zins, “Conceptual approaches for defining data, information, and knowledge,” *Journal of the American Society for Information Science and Technology*, vol. 58, no. 4, pp. 479–493, 2007.

[49] R. Ackoff, “From data to wisdom,” *Journal of applied systems analysis*, vol. 16, pp. 3–9, 1989.

[50] M. Hilbert and P. López, “The worlds technological capacity to store, communicate, and compute information,” *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[51] G. Piatetsky-Shapiro, “Knowledge discovery in databases: 10 years after,” *SIGKDD Explor. Newsl.*, vol. 1, no. 2, pp. 59–61, 2000.

[52] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[53] U. Fayyad, “Data mining and knowledge discovery: making sense out of data,” *IEEE Expert*, vol. 11, no. 5, pp. 20 –25, 1996.

[54] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., ser. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers In.c, 2005.

[55] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, 3rd ed. Academic Press, 2008.

[56] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. Tung, and H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[57] R. J. Duffin and A. C. Schaeffer, "A class of nonharmonic Fourier series," *Transactions of the American Mathematical Society*, vol. 72, no. 2, pp. pp. 341–366, 1952.

[58] J. Benedetto and S. Li, "The theory of multiresolution analysis frames and applications to filter banks," *Applied and Computational Harmonic Analysis*, vol. 5, no. 4, pp. 389 – 427, 1998.

[59] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[60] M. Thuillard, "A review of wavelet networks, wavenets, fuzzy wavenets and their applications," *Advances in Computational Intelligence and Learning*, pp. 43–60, 2002.

[61] D. Percival and A. Walden, *Wavelets Methods of Time Series Analysis*. Cambridge University Press, 2000.

[62] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, 1998.

[63] J. Li and A. Barron, "Mixture density estimation," *Advances in neural information processing systems*, vol. 12, pp. 279–285, 2000.

[64] S. Schwartz, "Estimation of probability density by an orthogonal series," *The Annals of Mathematical Statistics*, vol. 38, no. 4, pp. 1261–1265, 1967.

[65] J. Beirlant, L. Gyrfi, and G. Lugosi, "On the asymptotic normality of the l1- and l2-errors in histogram density estimation," *Canadian Journal of Statistics*, vol. 22, no. 3, pp. 309–318, 1994.

[66] R. Redner and H. Walker, "Mixture densities, maximum likelihood and the em algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[67] N. Céncov, "Evaluation of an unknown distribution density from observations," *Soviet Mathematics Doklady*, vol. 3, pp. 1559–1562, 1962.

[68] P. Hall, "On the rate of convergence of orthogonal series density estimators," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 1, pp. 115–122, 1986.

[69] E. Masry, "Probability density estimation from dependent observations using wavelets orthonormal bases," *Statistics and Probability Letters*, vol. 21, no. 3, pp. 181–194, 1994.

[70] D. Donoho, I. Johnstone, G. Kerkyacharian, and D. Picard, "Density estimation by wavelet thresholding," *The Annals of Statistics*, vol. 24, no. 2, pp. 508–539, 1996.

[71] D. Donoho and I. Johnstone, "Minimax estimation via wavelet shrinkage," *The Annals of Statistics*, vol. 26, no. 3, pp. 879–921, 1998.

[72] D. Herrick, G. Nason, and B. Silverman, "Some new methods for wavelet density estimation," *Sankhy : The Indian Journal of Statistics, Series A*, vol. 63, no. 3, pp. 394–411, 2001.

[73] A. Pinheiro and B. Vidakovic, "Estimating the square root of a density via compactly supported wavelets," *Computational Statistics & Data Analysis*, vol. 25, no. 4, pp. 399 – 415, 1997.

[74] B. Vidakovic, *Statistical Modeling by Wavelets*. Wiley New York, 1999.

[75] I. Daubechies and J. Lagarias, "Two-scale difference equations ii. local regularity, infinite products of matrices and fractals," *SIAM Journal on Mathematical Analysis*, vol. 23, pp. 1031–1079, 1992.

[76] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time

series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[77] M. Corduas and D. Piccolo, "Time series clustering and classification by the autoregressive metric," *Computational statistics & data analysis*, vol. 52, no. 4, pp. 1860–1872, 2008.

[78] K.J. and Astrom, "On the choice of sampling rates in parametric identification of time series," *Information Sciences*, vol. 1, no. 3, pp. 273 – 278, 1969.

[79] E. Keogh and M. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, ser. PADKK '00. London, UK: Springer-Verlag, 2000, pp. 122–133.

[80] F. Chung, T. Fu, R. Luk, and V. Ng, "Flexible time series pattern matching based on perceptually important points," in *International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*, 2001, pp. 1–7.

[81] T. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164 – 181, 2011.

[82] V. Megalooikonomou, G. Li, and Q. Wang, "A dimensionality reduction technique for efficient similarity analysis of time series databases," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, ser. CIKM '04. New York, NY, USA: ACM, 2004, pp. 160–161.

[83] R. Rengaswamy and V. Venkatasubramanian, "A syntactic pattern-recognition approach for process monitoring and fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 8, no. 1, pp. 35–51, 1995.

[84] G. Stockman and L. Kanal, "Problem reduction representation for the linguistic analysis of waveforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 3, pp. 287–298, 1983.

[85] D. Galati and M. Simaan, "Automatic decomposition of time series into step, ramp, and impulse primitives," *Pattern Recognition*, vol. 39, no. 11, pp. 2166–2174, 2006.

[86] J. Drakopoulos and B. Hayes-Roth, "tFPR: A fuzzy and structural pattern recognition system of multi-variate time-dependent pattern classes based on sigmoidal functions," *Fuzzy Sets and Systems*, vol. 99, no. 1, pp. 57–72, 1998.

[87] J. Lin, L. Keogh, E.and Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, pp. 107–144, 2007.

[88] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh, "iSAX 2.0: Indexing and mining one billion time series," in *IEEE International Conference on Data Mining, ICDM'10*, 2010, pp. 58–67.

[89] A. Camerra, J. Shieh, T. Palpanas, T. Rakthanmanon, and E. Keogh, "Beyond one billion time series: indexing and mining very large time series collections with iSAX2+," *Knowledge and Information Systems*, pp. 1–29, 2013.

[90] W. Zalewski, F. Silva, F. Wu, H. Lee, and A. Maletzke, "A symbolic representation method to preserve the characteristic slope of time series," in *Advances in Artificial Intelligence-SBIA 2012*. Springer, 2012, pp. 132–141.

[91] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 149–182, 2011.

[92] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '83.*, vol. 8, 1983, pp. 93–96.

[93] W. Chen, C. Hsieh, and E. Lai, "Robust speaker identification system based on wavelet transform and gaussian mixture model," in *Natural Language Processing IJCNLP 2004*, ser. Lecture Notes in Computer Science, K. Su, J. Tsujii, J. Lee, and O. Kwong, Eds. Springer Berlin - Heidelberg, 2005, vol. 3248, pp. 263–271.

[94] P. Nghia, P. Binh, N. Thai, N. Ha, and P. Kumsawat, "A robust wavelet-based text-independent speaker identification," in *IEEE ICCIMA '07*, vol. 2, 2007, pp. 219 –223.

[95] J. Gowdy and Z. Tufekci, "Mel-scaled discrete wavelet coefficients for speech recognition," in *IEEE ICASSP '00*, vol. 3, 2000, pp. 1351 –1354.

[96] E. S. García-Treviño and J. A. Barria, “Online wavelet-based density estimation for non-stationary streaming data,” *Computational Statistics & Data Analysis*, vol. 56, pp. 327–344, 2012.

[97] L. Golab and M. T. Ozsu, “Issues in data stream management,” *ACM Special Interest Group on Managment of Data (SIGMOD Rec.)*, vol. 32, no. 2, pp. 5–14, 2003.

[98] C. Aggarwal, “An introduction to data streams,” in *Data Streams (The Kluwer International Series on Advances in Database Systems)*, C. Aggarwal and A. Elmagarmid, Eds. Springer US, 2007, vol. 31, pp. 1–8.

[99] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” in *ACM SIGMOD-SIGACT-SIGART 2002.* Madison, Wisconsin: ACM, 2002, pp. 1–16.

[100] M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining data streams: a review,” *SIGMOD Rec.*, vol. 34, pp. 18–26, 2005.

[101] P. P. Rodrigues, J. Gama, and J. Pedroso, “Hierarchical clustering of time-series data streams,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 615–627, 2008.

[102] B. Dai, J. Huang, M. Yeh, and M. Chen, “Adaptive clustering for multiple evolving streams,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1166 –1180, 2006.

[103] J. Beringer and E. Hüllermeier, “Online clustering of parallel data streams,” *Data amp; Knowledge Engineering*, vol. 58, no. 2, pp. 180 – 204, 2006.

[104] S. Guha, N. Mishra, R. Motwani, and L. O’Callaghan, “Clustering data streams,” in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, 2000, pp. 359 –366.

[105] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O’Callaghan, “Clustering data streams: Theory and practice,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 515 – 528, 2003.

[106] C. Aggarwal, J. Han, J. Wang, and P. Yu, “A framework for clustering evolving data streams,” in *Proceedings of the 29th International Conference on Very Large Data Bases*, vol. 29. VLDB Endowment, 2003, pp. 81–92.

[107] ——, “On demand classification of data streams,” in *ACM SIGKDD 2004*. New York, NY, USA: ACM, 2004, pp. 503–508.

[108] C. Aggarwal and P. Yu, “On string classification in data streams,” in *ACM SIGKDD 2007*. New York, NY, USA: ACM, 2007, pp. 36–45.

[109] C. Faloutsos, “Indexing and mining streams,” in *ACM SIGMOD 2004*. New York, NY, USA: ACM, 2004, pp. 969–969.

[110] V. Gopalkrishnan, “Querying time-series streams,” in *EDBT 2008*. New York, NY, USA: ACM, 2008, pp. 547–558.

[111] R. Agrawal, C. Faloutsos, and A. Swami, “Efficient similarity search in sequence databases,” in *Foundations of Data Organization and Algorithms*, D. Lomet, Ed. Springer Berlin / Heidelberg, 1993, vol. 730, pp. 69–84.

[112] G. Cormode and M. Hadjieleftheriou, “Finding the frequent items in streams of data,” *Commun. ACM*, vol. 52, pp. 97–105, 2009.

[113] S. Tanbeer, C. Ahmed, B. Jeong, and Y. Lee, “Efficient frequent pattern mining over data streams,” in *Proceedings of the 17th ACM conference on Information and knowledge management*, ser. CIKM ’08. New York, NY, USA: ACM, 2008, pp. 1447–1448.

[114] J. Guo, P. Zhang, J. Tan, and L. Guo, “Mining frequent patterns across multiple data streams,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*, ser. CIKM ’11. New York, NY, USA: ACM, 2011, pp. 2325–2328.

[115] A. Bifet, “Adaptive learning and mining for data streams and frequent patterns,” *SIGKDD Explor. Newsl.*, vol. 11, pp. 55–56, 2009.

[116] D. Kifer, S. Ben-David, and J. Gehrke, “Detecting change in data streams,” in *VLDB 2004*. VLDB Endowment, 2004, pp. 180–191.

[117] D. Tran and K. Sattler, “On detection of changes in sensor data streams,” in *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, ser. MoMM ’11. New York, NY, USA: ACM, 2011, pp. 50–57.

[118] W. Fan, “Systematic data selection to mine concept-drifting data streams,” in *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’04. New York, NY, USA: ACM, 2004, pp. 128–137.

[119] C. Aggarwal, “A framework for diagnosing changes in evolving data streams,” in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’03. New York, NY, USA: ACM, 2003, pp. 575–586.

[120] A. Bulut and A. Singh, “SWAT: hierarchical stream summarization in large networks,” in *Data Engineering, 2003. Proceedings. 19th International Conference on*, 2003, pp. 303 – 314.

[121] ——, “A unified framework for monitoring data streams in real time,” in *ICDE 2005*, 2005, pp. 44 – 55.

[122] M. Datar, A. Gionis, P. Indyk, and R. Motwani, “Maintaining stream statistics over sliding windows: (extended abstract),” in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA ’02. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002, pp. 635–644.

[123] G. Cormode and M. Garofalakis, “Sketching probabilistic data streams,” in *ACM SIGMOD 2007*. New York, NY, USA: ACM, 2007, pp. 281–292.

[124] ——, “Sketching streams through the net: distributed approximate query tracking,” in *Proceedings of the 31st international conference on Very large data bases*, ser. VLDB ’05. VLDB Endowment, 2005, pp. 13–24.

[125] Y. Ogras and H. Ferhatosmanoglu, “Online summarization of dynamic time series data,” *The VLDB Journal*, vol. 15, pp. 84–98, 2006.

[126] B. Pan, U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, “Spatiotemporal summarization of traffic data streams,” in *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming*, ser. IWGS ’10. New York, NY, USA: ACM, 2010, pp. 4–10.

[127] L. Qiao, D. Agrawal, and A. El Abbadi, “Rhist: adaptive summarization over continuous data streams,” in *CIKM 2002*. New York, NY, USA: ACM, 2002, pp. 469–476.

[128] C. Pang, Q. Zhang, D. Hansen, and A. Maeder, "Unrestricted wavelet synopses under maximum error bound," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, ser. EDBT '09. New York, NY, USA: ACM, 2009, pp. 732–743.

[129] S. Guha and B. Harb, "Wavelet synopsis for data streams: minimizing non-euclidean error," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ser. KDD '05. New York, NY, USA: ACM, 2005, pp. 88–97.

[130] M. Garofalakis, J. Gehrke, and R. Rastogi, "Querying and mining data streams: you only get one look a tutorial," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '02. New York, NY, USA: ACM, 2002, pp. 635–635.

[131] Y. Yang, X. Wu, and X. Zhu, "Mining in anticipation for concept change: Proactive-reactive prediction in data streams," *Data Mining and Knowledge Discovery*, vol. 13, pp. 261–289, 2006.

[132] P. Domingos and G. Hulten, "A general method for scaling up machine learning algorithms and its application to clustering," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 106–113.

[133] B. Babcock, M. Datar, R. Motwani *et al.*, "Load shedding techniques for data stream systems," in *Proceedings of the 2003 Workshop on Management and Processing of Data Streams*, 2003.

[134] N. Tatbul, U. Çetintemel, S. Zdonik, M. Cherniack, and M. Stonebraker, "Load shedding in a data stream manager," in *Proceedings of the 29th international conference on Very large data bases - Volume 29*, ser. VLDB '03. VLDB Endowment, 2003, pp. 309–320.

[135] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58 – 75, 2005.

[136] C. Aggarwal and P. Yu, "A survey of synopsis construction in data streams," in *Data Streams*, ser. Advances in Database Systems, C. Aggarwal, Ed. Springer US, 2007, vol. 31, pp. 169–207.

[137] S. Guha, "Space efficiency in synopsis construction algorithms," in *Proceedings of the 31st international conference on Very large data bases*, ser. VLDB '05. VLDB Endowment, 2005, pp. 409–420.

[138] Y. Zhu and D. Shasha, "Statstream: statistical monitoring of thousands of data streams in real time," in *Proceedings of the 28th international conference on Very Large Data Bases*, ser. VLDB '02. VLDB Endowment, 2002, pp. 358–369.

[139] Y. Tao and D. Papadias, "Maintaining sliding window skylines on data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 377 – 391, 2006.

[140] M. Gaber, S. Krishnaswamy, and A. Zaslavsky, "Adaptive mining techniques for data streams using algorithm output granularity," in *The Australasian Data Mining Workshop*, 2003.

[141] M. Gaber, "Data stream mining using granularity-based approach," in *Foundations of Computational Intelligence*. Springer Berlin Heidelberg, 2009, vol. 6, pp. 47–66.

[142] A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss, "Surfing wavelets on streams: One-pass summaries for approximate aggregate queries," in *Proceedings of the 27th International Conference on Very Large Data Bases*, ser. VLDB '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 79–88.

[143] ——, "One-pass wavelet decompositions of data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 541 – 554, 2003.

[144] H. A., D. Habich, and M. Karnstedt, "Analyzing data streams by online dft," in *Proceedings of the fourth IWKDDS-2006*, 2006, pp. 67–76.

[145] B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan, "Maintaining variance and k-medians over data stream windows," in *ACM SIGMOD-SIGACT-SIGART 2003*. New York, NY, USA: ACM, 2003, pp. 234–243.

[146] X. Lu, T. Yang, Z. Liao, M. Elahi, W. Liu, and H. Wang, "Incremental outlier detection in data streams using local correlation integral," in *Proceedings of the*

2009 ACM symposium on Applied Computing, ser. SAC '09. New York, NY, USA: ACM, 2009, pp. 1520–1521.

[147] S. Papadimitriou, A. Brockwell, and C. Faloutsos, "Adaptive, unsupervised stream mining," *The VLDB Journal*, vol. 13, pp. 222–239, 2004.

[148] S. Guha, N. Koudas, and K. Shim, "Data-streams and histograms," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, ser. STOC '01. New York, NY, USA: ACM, 2001, pp. 471–475.

[149] H. Mousavi and C. Zaniolo, "Fast and accurate computation of equi-depth histograms over data streams," in *Proceedings of the 14th International Conference on Extending Database Technology*, ser. EDBT/ICDT '11. New York, NY, USA: ACM, 2011, pp. 69–80.

[150] Z. Li, H. Ma, and Y. Zhou, "A unifying method for outlier and change detection from data streams," in *Computational Intelligence and Security, 2006 International Conference on*, vol. 1, 2006, pp. 580 –585.

[151] E. Cohen, G. Cormode, and N. Duffield, "Structure-aware sampling on data streams," *SIGMETRICS Perform. Eval. Rev.*, vol. 39, pp. 157–168, 2011.

[152] T. Pavlidis, "Hierarchies in structural pattern recognition," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 737–744, 1979.

[153] C. Heinz and B. Seeger, "Wavelet density estimators over data streams," in *SAC '05: Proceedings of the 2005 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2005, pp. 578–579.

[154] N. Greggio, A. Bernardino, and J. Santos-Victor, "Unsupervised learning of finite gaussian mixture models (GMMs): A greedy approach," in *Informatics in Control, Automation and Robotics*, ser. Lecture Notes in Electrical Engineering, J. Cetto, J. Ferrier, and J. Filipe, Eds. Springer Berlin Heidelberg, 2011, vol. 89, pp. 105–120.

[155] E. Gardner, "Exponential smoothing: The state of the art," *Journal of Forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[156] Z. Hussain, A. Sadik, and P. OShea, *Digital signal processing: an introduction with MATLAB and applications*. Springer, 2011.

[157] K. Caudle and E. Wegman, “Nonparametric density estimation of streaming data using orthogonal series,” *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 3980 – 3986, 2009.

[158] K. A. Loparo, “Case western reserve university bearing data center,” http://www.eecs.case.edu/laboratory/bearing/, accessed on October-2010.

[159] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana, “The UCR time series classification/clustering homepage,” 2006, accessed on November-2011. [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_data/

[160] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, “Dimensionality reduction for fast similarity search in large time series databases,” *Knowledge and Information Systems*, vol. 3, pp. 263–286, 2001.

[161] J. Lin, E. Keogh, S. Lonardi, and B. 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*, ser. DMKD ’03. New York, NY, USA: ACM, 2003, pp. 2–11.

[162] M. Costa, A. Goldberger, and C. Peng, “Multiscale entropy analysis of complex physiologic time series,” *Physical Review Letters*, vol. 89, no. 6, 2002.

[163] K. Polat and S. Güne, “Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast fourier transform,” *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1017 – 1026, 2007.

[164] Y. Wu, D. Agrawal, and A. El Abbadi, “A comparison of dft and dwt based similarity search in time-series databases,” in *Proceedings of the ninth international conference on Information and knowledge management*, ser. CIKM ’00. New York, NY, USA: ACM, 2000, pp. 488–495.

[165] M. Misiti, Y. Misiti, G. Oppenheim, and J. Poggi, “Wavelet toolbox,” *The MathWorks Inc., Natick, MA*, 1996.

[166] L. Zhang, G. Xiong, H. Liu, H. Zou, and W. Guo, “Fault diagnosis based on optimized node entropy using lifting wavelet packet transform and genetic algorithms,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 224, no. 5, pp. 557–573, 2010.

[167] A. Jain, A. Ross, and S. Pankanti, "Biometrics: a tool for information security," *IEEE Transactions on Information, Forensics and Security*, vol. 1, no. 2, pp. 125 – 143, 2006.

[168] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[169] S. Z. Fatemian, F. Agrafioti, and D. Hatzinakos, "Heartid: Cardiac biometric recognition," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, 2010, pp. 1–5.

[170] F. Agrafioti, F. M. Bui, and D. Hatzinakos, "Medical biometrics: The perils of ignoring time dependency," in *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, 2009. BTAS '09*, 2009, pp. 1–6.

[171] F. Agrafioti, F. Bui, and D. Hatzinakos, "Medical biometrics in mobile health monitoring," *Security and Communication Networks*, vol. 4, no. 5, pp. 525–539, 2011.

[172] A. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010.

[173] J. Gertler, *Fault detection and diagnosis in engineering systems.* CRC, 1998.

[174] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers Chemical Engineering*, vol. 27, no. 3, pp. 327 – 346, 2003.

[175] S. Katipamula and M. Brambley, "Methods for fault detection, diagnostics, and prognostics for building systems a review, part I," *HVACR Research*, vol. 11, no. 1, pp. 3–25, 2005.

[176] ——, "Methods for fault detection, diagnostics, and prognostics for building systems a review, part II," *HVACR Research*, vol. 11, no. 2, pp. 169–187, 2005.

[177] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805 – 822, 1999.

[178] A. Sundaram, "An introduction to intrusion detection," *Crossroads*, vol. 2, no. 4, pp. 3–7, 1996.

[179] R. Bolton and D. Hand, “Statistical fraud detection: A review,” *Statistical Science*, vol. 17, no. 3, pp. pp. 235–249, 2002.

[180] Y. Kou, C. Lu, S. Sirwongwattana, and Y. Huang, “Survey of fraud detection techniques,” in *IEEE International Conference on Networking, Sensing and Control*, vol. 2, 2004, pp. 749 – 754.

[181] J. Chamberland and V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 407 – 416, 2003.

[182] V. Veeravalli and J. Chamberland, *Detection in sensor networks.* John Wiley & Sons, Ltd, 2007.

[183] L. Chen, L. Zou, and L. Tu, “A clustering algorithm for multiple data streams based on spectral component similarity,” *Information Sciences*, vol. 183, no. 1, pp. 35 – 47, 2012.

[184] Met Office, UK’s National Weather Service, accessed on November-2012. [Online]. Available: http://www.metoffice.gov.uk/

[185] R. Ogden, *Essential Wavelets for Statistical Applications and Data Analysis.* Birkhauser, 1997.

[186] M. Vannucci, “Nonparametric density estimation using wavelets,” Institute of Statistics and Decision Sciences, Duke University, Tech. Rep. 95-26, 1995. [Online]. Available: http://www.isds.duke.edu/

[187] EPDHK, Environmental Protection Department, Hong Kong, accessed on October-2012. [Online]. Available: http://www.epd-asg.gov.hk/