NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF PHYSICS

# GRAMMAR OF TIME
# TIME SERIES DATA MINING IN STEROIDS

## HOW TO TRANSLATE TIME?

JOHN VERY LONGNAME DOE

Master/BSc in Name of Previous Degree

DOCTORATE IN STUDY PROGRAM NAME

NOVA University Lisbon
month, year

DEPARTMENT OF PHYSICS

# GRAMMAR OF TIME
# TIME SERIES DATA MINING IN STEROIDS

## HOW TO TRANSLATE TIME?


**JOHN VERY LONGNAME DOE**

Master/BSc in Name of Previous Degree

**Adviser:** Mary Doe Adviser Name
*Full Professor, NOVA University Lisbon*

**Co-advisers:** John Doe Co-Adviser Name
*Associate Professor, NOVA University Lisbon*

John Doe other Co-Adviser Name
*Full Professor, NOVA University Lisbon*


**Examination Committee**

**Chair:** Name of the committee chairperson
*Full Professor, FCT-NOVA*

**Rapporteur:** Name of a rapporteur
*Associate Professor, Another University*

**Members:** Another member of the committee
*Full Professor, Another University*

Yet another member of the committee
*Assistant Professor, Another University*


DOCTORATE IN STUDY PROGRAM NAME
SPECIALIZATION IN SPECIALITY NAME

NOVA University Lisbon
month, year

**Grammar of Time Time Series Data Mining in Steroids**



This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.9.14) [**novathesis-manual**].

*Dedicatory lorem ipsum.*

# Acknowledgements

Acknowledgments are personal text and should be a free expression of the author.

However, without any intention of conditioning the form or content of this text, I would like to add that it usually starts with academic thanks (instructors, etc.); then institutional thanks (Research Center, Department, Faculty, University, FCT / MEC scholarships, etc.) and, finally, the personal ones (friends, family, etc.).

But I insist that there are no fixed rules for this text, and it must, above all, express what the author feels.

*"You cannot teach a man anything; you can only help him discover it in himself." (Galileo)*

# Abstract

Regardless of the language in which the dissertation is written, a summary is required in the same language as the main text and another summary in another language. It is assumed that the two languages in question are Portuguese and English.

The abstracts should appear first in the language of the main text and then in the other language. For example, if the dissertation is written in Portuguese the abstract in Portuguese will appear first, then the abstract in English, followed by the main text in Portuguese. If the dissertation is written in English, the abstract in English will appear first, then the abstract in Portuguese, followed by the main text in English.

In the LaTeX version, the NOVAthesis template will automatically order the two abstracts taking into account the language of the main text. You may change this behaviour by adding

```
\abstractorder(<MAIN_LANG>):={<LANG_1>,...,<LANG_N>}
```

to the customization area in the document preamble, e.g.,

```
\abstractorder(de):={de,en,it}
```

The abstracts should not exceed one page and, in a generic way, should answer the following questions (it is essential to adapt to the usual practices of your scientific area):

1. What is the problem?
2. Why is this problem interesting/challenging?
3. What is the proposed approach/solution?
4. What results (implications/consequences) from the solution?

**Keywords:** Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5, Keyword 6, Keyword 7, Keyword 8, Keyword 9

# Resumo

Independentemente da língua em que a dissertação esteja redigida, é necessário um resumo na mesma língua do texto principal e outro resumo noutra língua. Pressupõe-se que as duas línguas em questão sejam o português e o inglês.

Os resumos devem aparecer primeiro na língua do texto principal e depois na outra língua. Por exemplo, se a dissertação for redigida em português, o resumo em português aparecerá primeiro, seguido do resumo em inglês (*abstract*), seguido do texto principal em português. Se a dissertação for redigida em inglês, o resumo em inglês (*abstract* aparecerá primeiro, seguido do resumo em português, seguido do texto principal em inglês.

Na versão LaTeX o template NOVAthesis irá ordenar automaticamente os dois resumos tendo em consideração a língua do texto principal. É possível alterar este comportamento adicionando

```
\abstractorder(<MAIN_LANG>):={<LANG_1>,...,<LANG_N>}
```

à zona de customização no preâmbulo do documento, e.g.,

```
\abstractorder(de):={de,en,it}
```

Os resumos não devem ultrapassar uma página e, de forma genérica, devem responder às seguintes questões (é essencial adaptá-los às práticas habituais da sua área científica):

1. Qual é o problema?
2. Porque é que é um problema interessante/desafiante?
3. Qual é a proposta de abordagem/solução?
4. Quais são as consequências/resultados da solução proposta?

**Palavras-chave:** Palavra-chave 1, Palavra-chave 2, Palavra-chave 3, Palavra-chave 4

# Contents

# List of Figures

# List of Tables

# Glossary

This document is incomplete. The external file associated with the glossary 'main' (which should be called `template.gls`) hasn't been created.

Check the contents of the file `template.glo`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

If you don't want this glossary, add `nomain` to your package option list when you load `glossaries-extra.sty`. For example:

```
\usepackage[nomain]{glossaries-extra}
```

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "template"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "template"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

# Acronyms

This document is incomplete. The external file associated with the glossary 'acronym' (which should be called `template.acr`) hasn't been created.

Check the contents of the file `template.acn`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "template"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "template"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

# Symbols

This document is incomplete. The external file associated with the glossary 'symbols' (which should be called `template.sls`) hasn't been created.

Check the contents of the file `template.slo`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "template"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "template"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

# Chemical Symbols

This document is incomplete. The external file associated with the glossary 'chemical' (which should be called `template.chs`) hasn't been created.

Check the contents of the file `template.cho`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "template"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "template"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

# 1 Introduction

## 1.1 Time Series and Challenges of a Data-Driven Society

(**THIS SHOULD HAVE SOMETHING LIKE: WE BELIEVE THERE IS A LACK OF TOOLS THAT CAN PUT THE INTUITION OF THE USER INTO THE ANALYSIS PROCESS OF TIME SERIES. SEVERAL METHODS ARE ALREAY AVAILABLE BUT WE SHOULD MAKE THEM AVAILABLE AS FUNCTIONAL TOOLS. I HAVE NOT DONE THAT, BUT THIS WOULD BE MY ULTIMATE GOAL AND MOVE TOWARDS THAT.**

In recent years, the continuous increase in accessible wearable technology has contributed to a significant amount of data available. The continuous production of data from wearable devices through the usage of mobile phones, smartwatches, hearables, wristbands and other non-invasive wearable sensors has provided a valuable quantity of information. This data often comes as time series, being one of the most common data type in nature [**puttinghuman**]. As reported in *Tankovska et al.*, the wearable devices usage has more than doubled in the interval between 2016 and 2019, reaching 722 million [**tankovska_23_2020**] [**novathesis-manual**], leading to a large volume of time series data being gathered in all possible scenarios, by monitoring patients in healthcare institutions [**cpd_medical_1**, **cpd_medical_2**, **cpd_medical_3**, **cpd_medical_4**, **dataset6**, **dataset7**], tracking everyday activities of humans [**cpd_har_1**, **cpd_har_2**, **review_1**], recording machines in industrial processes or workers motion while performing their tasks [**antonio**, **sara**]. It has never been so easy to gather data about any aspect of our life, work, education, society or industry. Of course, having relevant information about a subject is beneficial, but the overwhelming amount of data brings tremendous challenges in the ability to save, process, analyze and retrieve interpretable and meaningful information from which we can act upon[**bigdata**]. Ultimately, it becomes even harder to have data well structured and labeled, considering that it is a sensitive and time consuming process, which complexity increases with data quantity. This is particularly problematic when developing machine learning applications (remember that Garbage-in Garbage-out - *GIGO*) [**roh2019survey**]. In the work of *Roh et al.* is mentioned that data scientists only rely on a small portion of the available datasets because it is too expensive to label all the data available [**roh2019survey**], and this is just an example of how much data can be unused.

We believe that we should do more with the data we have and for that, tools should be available to support and help analysts to accelerate the process of information retrieval from time series by making it more expressive and intuitive. In this thesis, we propose several novel methods that contribute to the information retrieval problematic. These methods are designed to help in (1) tell the story behind the time series by means of a visual representation that highlights its structure and organization, (2) make the search of patterns and events with more expressive queries and (3) move towards distance measures that are more human readable. These contributions are part of a general work born with this thesis, called *grammar of time*.

## 1.2 Linguistic Nature of Time Series

Time Series are a visual domain, from which humans can create a good intuition. It is inherent to our ability to see relevant structures and patterns. The reader can imagine a recurrent shape, such as the QRS complex of an electrocardiogram (ECG) signal that is interrupted by a noisy segment ( ). When interpreting this signal, we see that it has 3 representative segments and that the first is very similar to the third one. We could then represent the signal by *A B A*. Some shapes may be harder to distinguish, for instance, consider an accelerometer signal of a subject while walking and shifting to jogging regime ( ). Or a change in the shape of the arterial blood pressure (ABP) signal when there is a change in the subject's posture ( ). In both cases, the signal has 2 structures of a similar representative periodic pattern (A B).

This visual intuition is also very clear when a (non-)experience analyst is searching for specific shapes or patterns in time series. The reader may agree that scientists or other professionals often resort to describe the shape they are looking for. For instance, a physician may say "*I am searching for the T-wave, that represents the* `large peak`" ( ) or "*I am searching for the QRS complex, that looks like a* `sharp peak followed by a sharp valley` ( )".

This visual intuition also happens when analysts are trying to find differences between classes of signals. For instance, the following shapes (1) and (2) are different because "*shape 1 has a* `peak` *where shape 2 doesn't*".

Time series are carriers of information and the presence of a change in the regimes of a time series or the presence of a specific shape in a segment of a time series may be associated with a specific occurrence in the physical world and be attributed a meaning. This notion of structure and meaning is a good approximation of what represents the foundation of a language: grammar and meaning [**grammar**].

*Grammar* is generally defined as the book of rules that constitutes the structure of a language, and is modeled by the morphology and syntax [**grammar**]. The first is the structure of words, how these are built or morphed based on context, while the latter consists in organizing words in sequences to form larger linguistic units, such as sentences. Such as a language has morphological and syntax rules that represent its structural information, time series are also organized by a formal structure of ordered subsegments with specific morphological characteristics, organized to build larger segments. This introduces why this thesis is designated *grammar of time* and also introduces the reader further to the problematic that will be explored in this work. We will demonstrate how the developed solutions are helpful to several domains, with a special interest in showing how these can be helpful and meaningful in the context of occupational health.

## 1.3 Context and Relevance in Occupational Health

Work-related musculoskeletal disorders (WMSDs) prevail as the most common occupational disease in the European Union. These have a global impact on the well being of individuals and their quality of life in a range of working sectors [**Irastorza2010**], accounting for the second largest responsibility to disability worldwilde [**Luttmann2003**]. These are specially prevalent as upper limb or neck disorder (with 42% of all WMSDs cases reported) [**Seidel2019**] in several industry sectors, such as textile and automotive, where production processes with pre-defined motions and actions have a repetitive/cyclic nature. This has a negative impact on the risk to develop musculoskeletal disorders, with tremendous consequences to both workers and companies, leading to absenteism, early retirement and loss of productivity [**Trabalhadores**, **Varandas19**].

Several strategies have been implemented to identify, regulate and prevent occupational risk in manufacturing industries, such as (1) the inclusion of job rotation schedules, which promote a variation of the exposure throughout the working day [**jobrotation**] and (2) screening tools, for the assessment of occupational risk exposure, e.g. OCcupational Repetitive Action (OCRA), Rapid Upper Limb Assessment (RULA) or the Ergonomic Assessment WorkSheet (EAWS) [**ocra**, **rula**, **eaws**]. Nevertheless, these strategies are not optimal because they (1) are not automated, relying in observational methods and dedicated personal to inspect video records; (2) are not objective measures; (3) do not take into account differences among the worker's population, as anthropometric, age and experience variability; and (4) present single scores, being insufficient to explain the factors that contributed to this risk. With the advent of Industry 4.0, more companies are using modern strategies that follow digital solutions to provide direct and objective quantitative measures [**romero**]. An example of these incentives is the usage of wearable inertial devices for motion and posture tracking of workers.

Using inertial motion units (IMUs), time series can be collected, and relevant information can be directly measured, e.g. position and velocity of each body segment, postural angles between joints and gait parameters, making these important for ergonomics studies [**Caputo2019**, **Hang19**]. There are some limitations of using IMUs, mostly related with the long term bias (sensor drifting) arising from long acquisitions and the empirical process to fine tune sensor fusion techniques. Other systems can be used for motion capture, such as camera-based methods, but these rely in fixed setup of cameras, which is unmanageable in real industrial scenarios [**sara**].

**SE CALHAR, INCLUIR NA MOTIVACAO QUE EM ALGUMAS EXPERIENCIAS VERIFICAMOS DIFERENCAS ENTRE GRUPO ANTROPOMETRICOS**

The usage of time series in this context can play an important role in supporting the decision of ergonomists and other professionals of the industry. In order to develop systems that can use motion and postural data for direct risk assessment and reporting, several challenges arise in the time series data mining domain. For instance, considering the periodic nature of most manufacturing tasks, risk factors are calculated by working cycle. Therefore, methods should be developed to identify working cycles with some variability in their periodicity. In addition, real occupational scenario might have interruptions or changes in the working behavior, due to abrupt production stoppage, shift breaks or even changing to another workspace that has a different motion pattern.

Other questions also arise by ergonomists, such as "*can we find a pattern that has a sharp rise in the IMU from the arm?*" or "*when the worker is using a hand tool to make screwing, can we see a periodic pattern on the IMU from the hand?*, which represent specific patterns with a descriptive shape that can be seen on the signals and are specific of a task. These events can be relevant to study their precise impact on the worker's occupational exposure. Having ways to detect these patterns is of great relevance as well. In this study, we will show how the proposed solutions can have an impact in these problems, and how they contribute to provide relevant visual feedback for information retrieval from the occupational data and make the search of specific patterns more intuitive and expressive, even for non-experienced data analysts, such as ergonomists.

## 1.4 Research Questions

The previous sections introduced our main motivations related with the development of methods for information retrieval, provided context regarding the grammar of time framework and how the proposed solutions can have a significant contributions in occupational health assesment.

This project addresses all the range of topics of time series, from the moment data is acquired (*sensing*), processed for information retrieval (*analysis*) and how it is used to act upon (*decision making*). The main objectives are related with the development of methods for information retrieval (*analysis*) from time series for better decision making.

1. **Sensing** - Explore in depth the available technology to measure motion and postural variables in occupational scenarios for risk assessment. This will take into account which variables are associated with a risk, based on ergonomic standards. These measures are reaturned as time series, which are processed in the topic *analyzis*;

2. **Analyzis** - In this topic, several research paths are explored. **A -** study (1) how to perform structural information retrieval in time series for segmentation based on change points and periodic points and (2) how are the segments related. For this, we applied a feature-based transformation of the time series and similarity based measures to make a meaningful visual representation, from which the segmentation points can be extracted. **B -** explore symbolic representations of time series, studying how these can be used for more expressive and intuitive pattern search with the help of regular expressions and ultimately natural language. **C -** From the textual representation of time series, study if we can make a higher leveled distance measure, following standard text mining methods. The resulting outputs of these methods can be used to get relevant information to take better decisions, namely in the occupational domain;

3. **Decision Making -** Study meaningful summarization techniques and explore several real-life examples in how the developed methods can help analysts be more aware of the data and move towards a more *democratized* usage of data mining tools for information retrieval in time series.

With this work, we intend to contribubte to the state of the art in time series data mining with tools that provide more meaningful representations of time series, from which information can be retrieved with more meaning and at a higher level of abstraction, closer to the human intuition and visual interpretative abilities. This contributes towards more expressive methods and a democratization of these tools to accelerate the analysis process by experts in data mining and make non-experts capable of making high-level analysis.

## 1.5 Thesis Structure

This thesis provides a detailed description and explanation of the research work developed during the PhD program. It is organized in blablabla...

Figure X illustrates a guideline of the structure of this work, with a short description of each Chapter's content and description.

Chapter 1 introduced the main motivations, goals and context for the development of this thesis. Chapter 2 introduces theoretical concepts necessary to have a complete understanding of the work developed. It covers an introduction to motion and postural sensors used in occupational settings, time series, standard methods for its representation and analysis and text mining concepts. Chapter 3 presents the most recent works related with what we developed, namely in the topics of segmentation, summarization, pattern/event search and dictionnary based classification. On Chapter 4 we start describing the data we used, explaining its source for both acquired data and publicly available, for what it was used and how. It includes a detailed description of the protocol used to acquired workers' motion data in real industrial scenarios. The algorithm developed for time series structural information retrieval is explained in Chapter 5, while Chapter 6 covers the symbolic representation of time series. In this chapter is explained the exploratory path to use this novel representation in query search and classification tasks. Chapter 7 shows the application of the previous methods to an exaustive set of examples, namely from the occupational scenario, and major results are presented. In addition, this chapter also provide a general discussion about the usage of these methods for decision making. Finally, Chapter 8 gives an overall remark over the outcomes of this thesis and a reflection over the contributions that the developed methods have in making time series preparation and data mining more expressive, quicker and more practicle for an ever increasing number of data available. Each chapter will have a short introduction to situate and contextualize the reader.

# 2

# Theoretical Concepts

The content of this thesis is diverse and covers several different topics. Therefore, the reader will appreciate that we set the foundations that are necessary to fully capture the essence of this work. Fo this, we provide an introduction to each of the topics addressed, the global definitions and used notation in this work. We start by explaining occupational domain variables and corresponding sensors used to monitor these. The data of interest in this work is *time series* and the global definitions and notations are provided. Standard pre-processing methods, representation forms and distance measures are also explained. In this chapter, only global definitions will be made. Each further chapter will have additional and more contextualized definitions when needed.

## 2.1 Occupational Health Variables

### 2.1.1 Sensing the Physical World

- Bring an overview of the sensors used for a general set of things - motion, physiological, etc...

- These measure the physical world and retrieve physical changes in what they measure

- These changes can be related with something meaningful and relevant that occurred and can be seen on the data

- In this work, we apply the methods in all kinds of scenarios to show their agnosticism to domains, but focus our attention to a specific context where introducing sensing technology can have a strong impact.

### 2.1.2 Occupational Variables in the Industry

- Occupational variables that affect worker's health have long been studied and already defined in several screening tools from standard ergonomic guidelines. We can name EAWS, OCRA, RULA, etc...

- These worksheets are a reference for ergonomists in identifying the variables of interest to measure the risk of each activity performed by a worker.

- The multiple set of actions of a workstation can be analyzed

- Typically, these variables are related with motion and posture of body segments. Several scenarios are studied and variables extracted are frequency, intensity and duration of activity. Study specific activities, measure the risk based on vibration from machine or tools (https://www.cdc.gov/niosh/topics/ergonomics/ergoprimer/default.html)

- All these variables, being related with motion, should be studied

definition 1 - workstation

definition 3 - intensity, duration and frequency

definition 2 - vibration

- hand - is a special case. We should measure the vibration on the grip

### 2.1.3 Sensing Worker's Health

The type of variables that are required to perform a risk assessment are related with motion, posture and vibration. These are physical variables that can be quantified by means of inertial sensors, such as accelerometer, gyroscope and magnetometer. These three sensors are used together to compensate limitations of each other in error accumulation from sensor drifting.

- With these sensors, we can measure the orientation of body segments in terns of other body segments or standard body planes (sagital or frontal plane).

## 2.2 Time Series Fundamentals

The information gathered by sensor are physical quantities that vary with time. These are called *time series* and are the main topic of this work.

- **Definition 1 - Time Series (T) -** A time series is a sequence of real values ordered in time with length $n \in \mathbb{N}$: $T = (t_1, t_2, ..., t_n)$. Several domains of data rely in the acquisition of multiple time series from multiple axis of the same sensor (e.g. the 3-axis accelerometer) or from multiple sources (e.g. IMU as a fusion of three different sensors), creating a *multi-dimensional time series*.

- **Definition 2 - Multi-Dimensional T (MT) -** A $MT$ is a set of $k \in \mathbb{N}$ time series belonging to the same acquisition: $\{T_1, T_2, ..., T_k\}$.

  Segments of interest are often searched inside a *time series*. A segment is called a *subsequence*:

- **Definition 3 - Subsequence -** A *subsequence* is a segment of the time series with size $w \in \mathbb{N}$ and starting from a given position $i$ and ending at position $i$+$w$ from the $T$ or $MT$.

  A common strategy used in time series data mining is the moving window.

- **Definition 4 - Moving Window -** A *moving window* is a process of sliding along a time series $T$ to apply a specific method on each *subsequence* it hovers. The window has, such as the *subsequence* a predefined size $w \in \mathbb{N}$, which starts at a given position $i$ and ends at position $i$+$w$. The process is iterative and can be made overlapping windows or not. The next window will start at $i$+$o$, being $o$ the overlapping size.

With this process, each *subsequence* can be filtered, features can be extracted or distances can be measured. We will show several utilities of this technique further when introducing methods used to pre-process a raw time series.

Depending on the context and which conditions the data is gathered, the raw information can contain several sources of disturbance or should be transformed into another dimension to extract the information that matters. The set of tasks taken to prepare the *time series* to enhance information retrieval is called *pre-processing*.

The pre-processing steps we will discuss involve filtering, normalization and transformation.

### 2.2.1 Filtering

Time series have multiple sources of disturbance. This disturbance is usually called *noise* and is defined as an unwanted form of energy, but it can have multiple interpretations. It can be caused by internal sources inside a device, such as *white noise*, or be due to external sources, such as motion artifacts, wandering baseline, sensor detachment or the magnetic field from surrounding devices []. Any of these disturbances will affect the analysis stage and should be detected or removed.

Several methods can be used to reduce the influence of noise in the analysis. Standard filtering methods, such as low-pass, band-pass and high-pass filters can be used to reduce the presence of specific frequency bandwidths that are not relevant. There are many configurations for these types of filters, being one commonly used the *Butterworth* filter.

Another often used method that has the purpose of reducing the presence of noise and represents a variation of a low-pass filter is the smoothing technique. Several variations of this technique exist, being the simplest one a moving average, which uses a moving window, calculating the mean in each iteration.

Another type of disturbance on the data that is usually removed is a wandering baseline. An example typically occurs in ECG signals, where the respiration creates a wandering baseline on the signal. This type of disturbance has a very low frequency compared to the meaningful information on the data and can be removed by subtracting a *smoothed* version of the original data.

### 2.2.2 Normalization

Normalization of data is an important step in any data mining process. It is essential for data uniformization and scaling, while keeping the morphology and shape of the time series. Several methods can be used for this purpose, namely:

$$\overline{T} = \frac{T}{max(|T|)} \tag{2.1}$$

the normalized signal ($\overline{T}$) is scaled by the absolute maximum of $T$. It is the simplest approach to normalization and guarantees that values are scaled linearly and their modulus cannot be higher than 1.

A variation of this process is the normalization by the range of amplitudes, which is as follows:

$$\overline{T} = \frac{T - min(T)}{max(T) - min(T)} \tag{2.2}$$

here the signal $T$ is normalized to range between [0,1]. Another normalization method, called *z-normalization*, is very commonly used and relies on the distribution of its values:

$$\overline{T} = \frac{T - \mu_T}{\sigma_T} \tag{2.3}$$

where the time series $T$ is subtracted by its mean, $\mu_T$ and scaled by its standard deviation, $\sigma_T$. The resulting values represent how many standard deviations the signal is away from the mean.

### 2.2.3 Transformation

In information retrieval, data has often to be re-scaled, simplified, approximate or represented into another data type. Each can contribute in their own way to capture the most relevant and meaningful information, or discover a new type of information that once was hidden in the original data. Dozens of methods exist for time series representation, such as Singular Value Decomposition (SVD) or wavelet transform, but only a few will be explained.

One of the first and most well known techniques suggested for time series transformation was the Discrete Fourier Transform (DFT) fourier. The idea behind this concept is that any signal, of any complexity, is a decomposition of a finite number of sine waves. Each wave is represented by a complex number, known as the Fourier coefficient, transforming the signal from the time domain to the frequency domain [**fourier2**]. This transformation allows to see the signal in a different manner, highlighting which frequencies concentrate more or less energy. It unveils the presence of specific types of noise or artifacts, or periodic shapes. Figure **??** shows the transformation of a signal into the frequency domain.

Frequency properties are very relevant to characterize a time series, but others can also be used to get a full characterization of the signal. The process of feature extraction is also a transformation method commonly employed. It is performed by a moving window from which features are extracted. For each feature, $f$, a feature vector is computed.

- **Definition 5 - Feature Series -** FA *feature series*, $F$, is a feature representation of a time series with size m that depends on the overlap size $o \in \mathbb{N}$ of the sliding process: $m = \frac{n}{w-o}$. Considering the existence of MT, the *feature series* range from $f_{1,1}$ to $f_{k,m}$.

  When extracting more than one feature, these are grouped into a *feature matrix*.

- **Definition 5 - Feature Matrix -** $\mathbf{F}_M$*A* ***feature matrix***, $\mathbf{F}_M$**, is the set of** $r$ **features extracted for** $k$ **time series, with size** $r \times (k \times m)$**.**

Another common used transformation method to simplify a time series and reduce its dimension is the piecewise aggregate approximation(PAA) [**paa**]. The process is to keep the average of the $N$ equi-sized subsequences in which the original signal with length $n$ is segmented, which results in $\overline{T} = \overline{t_1}, \overline{t_2}, ..., \overline{t_N}$, such that [**paa**]

$$\overline{t_i} = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} t_j \tag{2.4}$$

An example is showed in Figure **??**. The resulting signal has reduced noise and size, while conserving its trend.

From this method, a new representation technique was born, transforming the signal from the numerical to the symbolic domain. It is called Symbolic Aggregate approXimation (SAX). This method applies PAA to a z-normalized time series and indexes a letter to each sample of the simplified signal based on the distribution of its amplitude values. The signal's amplitude values are separated in bins with equal probability. The number of bins is equal to the size of the *alphabet* chosen. Figure **??** shows an example of the signal transformed into a string with 3 letters in its alphabet. Such as the DFT, SAX opens doors to analyze time series in a completely different manner, profiting from the much acquired knowledge in text mining.

DEFINE A SAX OR SYMBOLIC TIME SERIES HERE

In this thesis we will use feature vectors for several purposes. We also propose a novel symbolic representation technique for time series that is used for expressive pattern search and classification. In order to perform search or classification, we have to be able to calculate the difference/similarity between two time series or *subsequences*.

### 2.2.4 Distance Measures

There is an exhaustive number of distance measure for time series, but two of the classical standard measures still provide state-of-the art results in most time series data mining tasks, namely the euclidean distance (ED) and the dynamic time warping (DTW).

The ED is the most straightforward distance measure for time series. Let us consider two time series, $Q$ and $C$, of length $n$, so that

$Q = q_1, q_2, ..., q_i, ..., q_n$

$C = c_1, c_2, ..., c_i, ..., c_n$

The distance between these two time series under the ED is:

$$ED(Q,C) = \sqrt{\Sigma_i^n = 1(q_i - c_i)^2} \tag{2.5}$$

which represents the square root of the sum of the squared amplitude differences between the samples of each signal. Although the distance measure is simple to compute, it is highly susceptible to typical distortions on time series. When using ED, these distortions must be removed, otherwise, other methods, invariant to these distortions, should be used. Examples of distortions are the amplitude and offset distortion, phase distortion, and local scaling ("warping") distortion. The first can be compensated by the z-normalized ED:

$$z\_ED(Q,C) = \sqrt{2m(1 - \frac{\Sigma_{i=1}^m Q_i C_i - m\mu_Q\mu_C}{m\sigma_Q\sigma_C})} \tag{2.6}$$

where $\mu_Q$ and $\mu_C$ are the mean of the time series pair and $\sigma_Q$ and $\sigma_C$ are the standard deviation.

The *warping* distortion can be solved with an elastic measure. For this purpose, DTW is typically used.

The DTW distance measures the alignment between two time series. Let us consider two time series, $Q$ and $C$, of length $n$ and $m$, respectively:

$Q = q_1, q_2, ..., q_i, ..., q_n$
$C = c_1, c_2, ..., c_j, ..., c_m$

The alignment is measured by means of a distance matrix with size $n$-by-$m$, where the $(i^{th}, j^{th})$ cell of the matrix contains the $d(q_i, c_j)$ between the two points $q_i$ and $c_j$, being $d = (q_i - c_j)^2$ **[dtw]**. Figure **??** shows an example of a distance matrix between two time series. The matrix fully describes the difference between the two time series and maps where these align. The mapping is made by a warping path, $W$, that represent the set of matrix cells that minimize the warping cost, also defined as the cumulative distance of these cells [**dtw**]

$$W = w_1, w_2, ..., w_k, ..., w_K; \qquad max(m,n) \leq K < m + n + 1 \tag{2.7}$$

$$DTW(Q,C) = min\sqrt{\Sigma_{k=1}^K w_k} \tag{2.8}$$

The cumulative distance $\gamma(i,j)$ is calculated as $d(q_i, c_j)$ of the current cell added to the minimum distance adjacent to that cell:

$$\gamma(i,j) = d(q_i, c_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{2.9}$$

When two time series with the same length have a linear warping path, such that $w_k = (i,j)_k, i = j = k$, we have a special case of the ED. DTW has a time and space complexity of $O(nm)$ while the ED has linear complexity ($O(n)$).

A different type of distance measure is also used to cope with complexity invariance. This distance uses a complexity correction factor ($CF$) with an existing distance measure, such as ED [**complexity**]:

$$CD(Q,C) = ED(Q,C)(Q,C) \tag{2.10}$$

The $CF$ is defined as [**complexity**]:

$$CF = \frac{max\{CE(Q), CE(C)\}}{min\{CE(Q), CE(C)\}} \tag{2.11}$$

where $CE$ represents the complexity estimate of a time series. This estimate is calculated based on the intuition that if we could "stretch" a time series until it becomes a straight line, this line would be as long as the complexity of the signal. It can be computed as the sum of the $n-th$ discrete differences along the time series[**complexity**]:

$$CE(Q) = \sqrt{\Sigma_{i=1}^{n-1}(q_i - q_{i+1})^2} \tag{2.12}$$

These distance measures are performed on the original representation domain of time series. As we showed above, other representation techniques can be employed, creating opportunities for other types of approaches. In this work, we explore other representation techniques to create novel ways of exploring time series. Then, we find that the reader will appreciate that we describe other distance measures employed, namely in the feature-based domain and symbolic-based domain.

As mentioned, a feature series $F$ can be computed from the original time series to represent it based on a specific feature. If the size of the *moving window* is equal to the size of the time series, than $F$ is represented by a single value. Otherwise, each *subsequence* highlighted by the *moving window* is characterized by the feature and the $F$ is computed as an array. When multiple features are extracted, each *subsequence* is characterized by a set of features, creating a feature vector $\vec{f}$ with $r$ feature values. Vector based distance measures can be used with these feature vectors to compare different time series or *subsequences*. There are several vector-based distance measures, including the already mentioned euclidean distance or the manhattan distance, but we will only describe the cosine similarity/distance.

The cosine similarity is a measure of the angle between two vectors determining if these are pointing in the same direction. Consider two feature vectors $\vec{f_a}$ and $\vec{f_B}$. Their cosine similarity is computed as the normalized dot product:

$$CS = \frac{\vec{f_A} \cdot \vec{f_B}}{||\vec{f_A}||||\vec{f_B}||} \tag{2.13}$$

being $||\vec{f_A}||$ and $||\vec{f_B}||$ the euclidean norm of each feature vector, defined as $\sqrt{\Sigma_{i=1}^{r} f_{Ai}}$ and $\sqrt{\Sigma_{i=1}^{r} f_{Bi}}$, respectively.

This feature set characterizes the TS (MTS) in several dimensions that range from spectral, temporal or statistical. This characterization is used to compute the *self-similarity matrix*.**Definition 6 - Self-Simialrity Matrix (SSM) -** An *SSM* is a pairwise distance matrix between each *subsequence* of the *TS*. In this work, it is computed by simply calculating the dot product between the $F_M$ and its transpose. The *SSM* reveals several meaningful structures that indicate the presence of *events*. **Definition 7 - Event** An *event* is an instant in time $e$ that indicates the presence of a relevant occurrence in the TS. This instant is one sample of the TS. Multiple *events* segment the TS into several *subsequences* of different lengths. Therefore, event detection is often considered TS segmentation [**cpd_alan**]. In this work, we highlight two categories of event significance, based on the type of *event* that we search, namely *change point event* and *periodic event* **Definition 8 - Change Point Event -** A *change point event* indicates a significant change between one instant defined by *subsequence i* and *subsequence i+1*, segmented the time series. These events can be computed with the *novelty function*. **Definition 9 - Periodic Event -** A *periodic event* segments the periods of a cyclic time series into cyclic *subsequences*. These events can be computed with the *similarity function*.

This work proposes a structure-based method to perform classification of time series. Using a textual representation of time series. It promotes the usage of text-based queries for the description of the signal and textually transcribes its dynamics. This section explains what is a time series, its relationship with the linguistic transformation and how this solution fits within the problematic.

### 2.2.5 Global Definitions

We start by defining what is a *time series*:

***Definition 2.1*** A Time Series is an ordered sequence of $n \in \mathbb{N}$ real-valued samples, indexed by time.

$$ts = (ts_1, \cdots, ts_n)$$

In this work we transcribe structures of the time series to text, which correspond to *subsequences*.

***Definition 2.2*** A Time Series *subsequence* is a segment of a time series with an ordered sequence of $s \in \mathbb{N}$ real-valued samples, indexed by time, with $length_s < n$:

$$sts = (sts_1, \cdots, sts_s)$$

These *subsequences* are associated to a word or sequence of words, which can be defined as a *pattern*.

***Definition 2.3*** A Pattern is a *subsequence* or a combination of *subsequences* of a time series with a specific morphological representation or shape.

These *patterns* are characterized by *characters* and *words*, which can be derived by means of *SSTS* queries.

### 2.2.6 Time Series Textual Abstraction

The proposed method, conceptually developed based on text mining techniques, abstracts how a time series can be structured in a linguistic representation, similar to how the human would describe a time series with words. In order to introduce the reader with this abstraction and representation, we explain how we use *SSTS* to make this abstraction.

The transformation from the numerical domain to the textual domain is made using *SSTS* [**ssts**]. This method uses three steps to perform a query search on the time series and finding the corresponding pattern. The steps include (1) the pre-processing; (2) the symbolic connotation and (3) the search:

- Pre-processing: prepare the signal for the translation into the textual domain, removing noise or any disturbance in the signal that affects the pattern search;
- Connotation: transforms each sample of the time series into a character by extracting properties of the signal that are based on a conversion rule either defined by the user, or pre-defined in our vocabulary;
- Search: regular expression query that is matched on the textual pattern and corresponds to a *pattern* on the *time series*.

An example of the detection of shapes with the help of *SSTS* in a set of time series is made in Figure **??**. The example shows the potential of this mechanism to create the description made in Figure **??**.

We therefore define *character*, *word* and *sentence*, based on the concepts mentioned above.

***Definition 2.4*** A *character* is an unit symbolic element. In our work, we use the **connotation** step from *SSTS* to transcribe the time series into a sequence of *characters* based on a property. The mentioned properties, corresponding characters used to translate the time series and associated words are the following:
***amplitude:*** binary representation of the amplitude based on a threshold. Character ***0*** means the sample is below the threshold while ***1*** means it is above it.
***first derivative:*** estimated slope that a sample belongs to, with a threshold to limit flat areas. Characters given to each sample are ***p*** when slope is positive and higher than the

Figures/SSTS_example.png

Figure 2.1: (Top) Using SSTS to detect the rising stage of a time series. Each step of the process is written described as follows: (1) pre-processing: *Sm* is the function *Smooth* with a window size of 25 samples; (2) connotation: *D1*, indicates the first derivate, from which each sample is converted to *z* - Flat, *p* - rising and *n* falling; (3) search - regular expression *p+* searches for all sequences with 1 or more *p* characters. (Bottom) Example of sentence generation. Using the other search queries (*p+*, *n+*, *z+*), we can find the derivative patterns and convert it into ordered words.

threshold, ***n*** when negative and higher than the threshold, while ***z*** when the slope is below the threshold.

***height of a slope:*** Describes the slope in terms of height, that is, how high is a slope, either if positive or negative. The characters are case sensitive, being ***r*** for a low rise and ***R*** for a high rise. The same for falling (***F*** or ***f***).

***slope speed:*** the amplitude of the first derivative, characterizing quick (***Q***) and slow (***q***) samples of the signal.

***second derivative:*** second derivative of a signal to indicate concavity. When positive, concavity is convex (***C***), while when negative, it is concave (***D***).

The transcription results in sequences of *characters*. Specific *characters* sequences are translated into *words*.

***Definition 2.5*** A *word* is a sequence of *characters*. *SSTS* has the **search** step which provides a way of extracting specific patterns with regular expression queries. We pre-defined a set of search queries that correspond to specific words, such as *rise*, *fall*, *peak*, *valley*, etc.... These words can be associated with a single character (e.g. rise = ***p+***) or a combination of them (e.g. peak = ***p+[z]n+***). Examples of common matches are presented in Figure.

This set of words belong to a *vocabulary*.

***Definition 2.6*** The *vocabulary* comprehends the set of words available in a pre-defined set of *SSTS* queries.

The *words* used can be ordered to form a *sentence*.

***Definition 2.7*** A *sentence* is a set of *words* or tokens organized sequentially. In this work we create multiple sentences based on the types of *connotation* methods used to transcribe the time series. In Figure **??**.Bottom, a time series is translated into the *sentence* ***Flat Rise Fall Rise Flat*** based on a set of *SSTS* queries from the derivative connotation (***p+, z+*** and ***n+***). For each time series, one document is generated.

*Sentences* can be added to a *document*.

***Definition 2.8*** A *document* is a piece of text with a collection of words or tokens that are used to build sentences. It can be made of only one sentence or multiple ones. In this work, a *time series document* will have multiple sentences as groups of describing patterns. Finally, the higher hierarchy in textual information is the *corpus*.

***Definition 2.9*** The *corpus* is a collection of text material (group of documents). It represents the higher level of textual information. This collection is typically annotated and used for machine learning tasks. In this case, a corpus will be represented by the set of documents that describe a time series dataset.

Since this work follows the steps of *NLP* strategies for document classification, we will define *Bag of Words* (BoW).

***Definition 2.10*** A BoW is a feature matrix representation of a corpus, being the feature the number of occurrences of each word, the term-frequency (*tf*):

$$tf_{t,d} = \frac{f_{t,d}}{\sum\limits_{t' \in d} f_{t',d}}, \tag{2.14}$$

being $t$ the term that exists in a document, $d$ the document, $t'$ the term that belongs to document $d$.

We use the BoW to vectorize the textual representation of each time series.

Most of the times, *n-grams* are used in addition to single words.

***Definition 2.11*** *N-grams* are a span of followed words that are counted in the BoW. It gives more context to words that are frequently followed by a specific word. As with time series we have temporal order of occurrences, this feature is very important. An example of *2-gram* from the sentence *Rise Flat Fall* would be *Rise Flat* and *Flat Fall*.

## 2.3 Sensing Human Posture, Motion and Physiology

## 2.4 Linguistic Nature of Time Series

# 3

# State of the Art

State of the art in the topics mentioned previously as well as what has been done considering the advances in time series data mining.

## 3.1 Information Retrieval from Time Series

### 3.1.1 Event Detection

Most of the works available in event detection are focused in change point detection or segmentation. The found strategies are categorized based on (1) their ability to be used online or offline, (2) being univariate or multivariate, (3) based on a model or non-parametric and (4) being unsupervised or supervised [**cpd_alan**, **review_1**, **review_2**]. Regarding supervised methods, there are multi-class, binary and virtual classifiers, optimized for the purpose of detecting change points [**review_cpd_1**]. The advantage of supervised methods is to not only detect the change point, but give the nature of the change as well. Another example uses neural networks with transfer learning for segmentation [**pedromatias**]. However, supervised methods rely in very brittle training sets and class imbalance, since there are more in-state sequences than change point sequences [**review_cpd_1**]. Additionally, a problem reported by [**cpd_alan**] is that most algorithms were validating the performance of their algorithms in synthetic data, which given the nature of the application was not optimal. In that sense, a benchmark is now available for change point detection [**cpd_alan**], where methods can be compared on real-data. The proposed work uses this benchmark to compare itself with other non-supervised and offline methods.

Existing non-supervised methods include older but with state of the art performance in change point detection, such as the *Baysian Online* method (BOCPD) [**bocpd**], the *binary segmentation* (BINSEG) method [**binseg**] and the *segmentation neighborhoods* (SEGNEIGH) method [**segneigh**]. These methods have been reported successful in several domains [**cpd_alan**], however, the BOCPD only achieved good results when parameters were hypertuned, and the BINSEG and SEGNEIGH are not used in multidimensional domains. In addition, these methods are not reported to cope with a multi-time scale change [**cpd_alan**]. An available repository provides an implementation of some of these offline methods [**review_2**], but these lack a visual output that might give the user an intuition over where a change point might be.

Another method, called FLOSS [**eamonn1**], relies in searching change points based on the nearest neighbors of subsequences, being very successful in real data domains. As it searches for nearest neighbors, the similarity between segments might be compared and used for summarization, but nothing is reported regarding multi-dimensional time series.

The ?? has been used for change point detection in the audio domain, based on a feature representation of the audio signal [**MuellerZ19_FMP_ISMIR**]. The advantage of using the ?? is the amount of information it provides for a specific time scale. In this work, we profit from these ideas applied in the audio domain, but extend its usage to other time series domains. The tool we propose can be used to detect events with context, associating the estimated events with patterns, (dis)similarities, periodicity and novelty. In addition, if being able to extract the information available in the ??, this tool can be extended to summarization tasks. Finally, although the search mechanism is based on a specific time scale, the process can be made recursive to perform multi-time scale searches recursively.

The proposed method highlights itself for being domain agnostic, work with both uni and multidimensional time series, give events with context by means of the visual information available, but also by the similarity measures in the matrix, that help in associating an event as a change or a periodic segment, and how similar are the segmented subsequences. It is unsupervised and works offline. It can be extended to work in multi-time scale problems with a special interest in time series summarization. We will demonstrate in this work how this method can bring novelty to the problematic of event detection, with a direct application to labelling and time series summarization.

The problems regarded in this work involve essentially the identification of cyclic information and anomalies. Typically, algorithms developed for these purposes may resort to (1) supervised machine learning (ML) methods, which require a certain level of annotation beforehand and (2) unsupervised methods, which are based on the similarity analysis of the signals or their features, without any prior information. Several methods found, employed in the analysis of inertial data, are used in the context of human activity recognition (HAR). The list of supervised ML methods is extensive and promising works are found to achieve this purpose. The application of neural networks [**Lara2013**], hidden Markov models [**Zhu2009**], decision trees [**Jatoba2008**], bayesian networks [**Jatoba2008**], and semi-automatic process [**duarte1**], among others, are algorithms capable of detecting and classifying various human actions. Nonetheless, most of the work done in this context only looks to identify previously defined actions like lying, standing, sitting down, move upstairs, etc., that might not be cyclic and rely on a significant amount of labelled data.

Several works that use unsupervised methods for the identification of cyclic information and anomalies are also found. The most simple method of cycle detection is the use of point references on the workplace to describe when a cycle starts and ends. Which is usually considered a system subject to flaws with a requirement for further adjustments steps [**Bauters2014**, **Bauters2018**]. Other more reliable alternatives analyze features of the signal and search for periodic motion in those. An automated algorithm of segmentation was able to separate complex and multidimensional data into smaller segments that can be described through harmonic models. This algorithm revealed to be significantly useful to identify cyclic movement without any *a priori* knowledge of the input data, using a combination of a recursive least squares segmentation algorithm, a model fitting of damped harmonics, and in the end, a clustering analysis to classify the events [**Lu2004**, **Lu2003**]. The usage of features is of great relevance in unsupervised works, and methods are found to select adequate features for detection and classification tasks, such as in [**machado2015**]. Another example is the use of four-pass UKF (unscented Kalman filter) to produce an unified model with kinematic parameters. These may then be segmented by analyzing the parameter's zero crossing velocity and in the end uses a clustering algorithm to identify repetitive segments [**Wang2015a**].

Other methods rely on a self-similarity approach, namely [**neuza**], where cyclic information is segmented by searching for minimums, in the convolution of a segment of the signal with itself. The *Matrix Profile (MP)*, which is a method that compares all sub-sequences of a given time series with themselves through an euclidean distance, has also revealed promising results. In the end, it returns the minimum value distance for each segment, highlighting the moments of the time series which are similar within themselves [**Yeh2018**]. Additionally, autocorrelation revealed itself an useful tool, as the search over maximum values can infer the cyclic nature of the data [**Bauters2014**]. Finally, for anomaly detection in industrial scenarios, an interesting work applies an unsupervised method based on the clustering of time series segments to detect the execution of improper movements [**duarte2**].

The following work is inspired over an algorithm for the detection of musical structures on audio signals [**Foote2000**, **audiolabs1**, **audiolabs2**] by means of a *Self-Similarity Matrix* (*SSM*). This sort of analysis of self-similarity to collect information about the periodicity has also been performed over video datasets. This type of analysis usually consists on a framework where a Fourier analysis is performed on an *SSM* to characterize and highlight the periodicity of the data from the video [**Cutler2002**, **Cutler2000**, **Cutler1999**].

### 3.1.2 Summarization

Very few strategies are found to make compact and meaningful representation of time series. The works that can be highlighted refer to time *snippets* and time series *bitmaps* [**snippets**, **bitmap**]. The first highlights the limitation of current methods in providing a satisfactory solution to time series *summarization*. It proposes a method that is able to segment the *k* most *representative* sub sequences of a time series, and use these elements as the summary. This strategy answers several of the discussed demands aforementioned in Section **??**, namely the segmentation and similarity. Regarding the time series *bitmap* representation, the strategy is able to provide a coded bitmap with information on cluster, Figures/keogh_examples.pdf

Figure 3.1: Strategies for time series summary found on the literature. These images are taken from the works from [**snippets**, **bitmap**]

anomaly and other regularities on data collection. These bitmaps were used as folder icons, and also answer several of the aforementioned characteristics, such as *similarity* and *events*. An example of both strategies can be seen on Figure **??**.

Time series *shapelets* are also a method that could provide interesting results. However, the strategy is *supervised*, and the point of the proposed method is to have *no apriori* knowledge about the structure of the data, except the time scale in which the summarization is performed.

Other interesting strategies provide a transformation of time series into text and could be used for time series summarization, but are not able to suitably summarize a time series from the textual representation [**ssts**, **sax**].

Strategies that are typically used to present information in a compact way are found in several domains. In text analysis, for instance, the relationship between repeating sequences is illustrated with arc diagrams [**bitmap**, **arcplots**]. These show where repeating sequences occur in a very concise way. This has a range of applications that include, for example text and DNA sequence analysis.

One domain that has a particular relevance in data visualization is genomics. Graphical genome maps are found to concatenate a significant amount of information in a very compact way. Genome features and sequence characteristics are assessed with this visual strategy. An example can be found on Figure **??**. This visualization strategy can provide increasing circular layers of information. Although we are used to look at time series from left to right, a circular representation can have benefits to concatenate the information we want to include.

In the musical domain, strategies have also been developed that summarize audio time series with segmentation techniques. One of the strategies that is common to be used involves detecting novelty instances on a similarity matrix representation of the audio signal, called *Self-Similarity Matrix* (SSM). This data structure provides a significant range of information that can be used to retrieve structural information, such as block and periodic structures [**fmp1**, **fmp2**, **audiolabs1**, **audiolabs2**]. This method will inspire our visualization strategy, which will be explained further.

### 3.1.3 Classification

Current available methods for time series classification are categorised as shape-based and structure-based. Existing approaches until the last decade were focused in shape-based similarity methods, while during the mid 2010's, methods that would seek the analysis of higher-level features started to be developed [**Keogh2004**].

Shape-based methods focus their attention in performing local comparisons between time series. Examples of well-known methods are the Euclidean distance (ED) or Dynamic Time Warping (DTW) [**jlin2013**]. Although both work well with short-length time series, the first has the inconvenient of needing time series with the same length, while also being sensitive to time misalignments. The latter is able to counteract this problem by means of determining the best alignment between two time series [**Keogh2004**, **jlin2013**]. These distance measures are usually combined with a k-Nearest Neighbour (k-NN) classifier to solve TSC tasks. The limitations of these techniques come with problems that include the presence of noise or long time series with characteristic sub-structures [**BOSS**].

In the other end, structure-based methods rely on broader aspects of time series such as the presence of specific morphological structures or patterns, being useful to classify long and noisy time series [**BOSS**]. Dictionary based methods fit into this category and have recently been used with great success. These techniques rely in a transformation of the time series into a symbolic feature vectors by means of a specific method, such as the *Symbolic Aggregate approXimation* (SAX) [**SAX**] or the *Symbolic Fourier Approximation* (SFA) [**SFA**]. The first approach proposed for TSC with symbolic representations was the work of Jessica Lin *et. al* with the *Bag of Patterns* (*BoP*)[**jlin2013**]. Further proposed methods were conceptually inspired on the *BoP*, using the same reasoning. Techniques such as *Bag of SFA Symbols* (BOSS) and Word ExtrAction for time SEries cLassification (WEASEL), from the same authors, use a similar reasoning but employ the SFA instead [**BOSS**, **weasle**].

Using syntactic methods has already been successful for several time series data mining tasks, mostly related with query search and classification. Besides, these methods, being dictionary-based, can be used to show similarity between subsequences by looking into the distribution of word counts. However, current methods rely mostly in incomprehensible sets of characters, such as *aaa*, which are hard to associate with a specific subsequence of the time series, therefore providing limited interpretability. In this work, we propose a method that literally translates the time series into sentences, such as that if a human was to describe a time series with text, it should be possible to separate these time series with the written words. We have seen natural language being used to include the human in the loop for more intuitive and meaningful query searches in time series [**hil_naturallanguage**]. Such as with SSTS, the purpose is to increase the expressiveness. This kind of descriptive power can be used to provide more intuitive feedback and increase interpretability to understand why a time series is different than others.

There is an existing method that is capable of providing visual interpretability of differences between time series, which is a structure-based method called *shapelets* [**shapelets**]. Shapelets are representative subsequences of the time series, which characterize a specific class. The advantage of this method is the higher interpretability because relevant shapes from the class can be highlighted [**shapelets**].

All the mentioned methods are a reference in TSC tasks with innovative concepts that merge ideas from the text-mining domain into TSC domain. One of the advantages of structure-based methods that rely in a dictionary-based concept is to use the words extracted as an interpretable model to differentiate time series. The histogram of words generated gives the user an understanding of which patterns better represent the time series and give an intuition over patterns that differ between classes of time series. This provides a feedback and explanation over why a class is different than the other. However, dictionaries can be confusing, and the words generated are not intuitively associated with the patterns these represent in the time series. One method that went beyond the previously mentioned methods in that aspect is the SAX-Vector Space Model (SAX-VSM). This method used a weighted word vector representation of the time series and showed which are the relevant words for the classification process and what patterns these represent in the time series, demonstrating that the classification process can be interpretable by measuring the importance of the patterns found for each class of signals [**sax_vsm**].

The proposed method is built upon the same ideas as the BoP method but uses the *SSTS* Tool to promote the inclusion of the human reasoning in the classification process and provide more interpretable representations, as inspired by the work of SAX-VSM.

The method has been conceptually designed focusing in providing a solution that copes with (1) enabling the human intuition in the classification process, (2) be invariant to size, (3) have awareness of the order at which structures appear on the time series, (4) be domain agnostic, (5) have a flexible pre-processing to increase the representational power and (6) increase the readability. This method brings novelty by using literal natural language sentences to perform classification of time series, which can be customized by an analyst and moves towards a more readable output on distinguishing time series both visually and with keywords.

### 3.1.4 Search by Query

There is a large literature on time series similarity search, see [26] and the references therein. However, in most cases it is assumed that the query comes from a downstream algorithm, not a human. As such, there has been relatively little attention paid to the ability of humans to formulate meaningful queries. In principle one could do "query-by-sketching" and invite the user to draw the pattern she is interested in finding [15,16]. The recent "Qetch" system is a prominent example of this approach [15]. However, there are two possible limitations to such an approach: First, it is not clear that most people have the ability to sketch their query. For example, many people cannot even draw an accurate circle [25]. Secondly, as Figure 1 hinted at, classic distance measures may be too literal and limited in expressiveness to retrieve the desired pattern. As a simple example, suppose that a user wishes to retrieve all highly symmetric patterns. There is simply no way to do this with Euclidean distance or similar distance measures. Other researchers have noted these issues and proposed more flexible queries languages for time series. For example, the SDL (shape definition language) of [11]- allows the user to formulate "blurred" queries. However, we believe that most such systems are not accessible for the typical user. For example, in our proposed system, a 3-point-turn can be successfully queried by noting that the surge axis will exhibit three consecutive "bumps" and formulating the query Surge: [peak peak peak]. In contrast, SDL would require: (Shape triplespeak (width ht1 ht2 ht3) (in width (in order spike (ht1 ht2 ht3) spike (ht1 ht2 ht3)))). Several similar query systems based on regular expressions or SQL-like languages have been proposed, but none seem suitable for general use [17,20]. There have also been a handful of other attempts at natural language querying for time series [6,7]. None of these works seem to have been adopted by practitioners. We feel that this is because they probably suffer from too broad an ambition, proposing completely domain independent search. While domain independence is a worthy ambition (and our eventual research goal), it is clearly challenging. Even the word "spike" can have a different meaning for neuroscientists, economists, epidemiologists, and astronomers. In this work we take advantage of the fact that driving is a familiar, even quotidian, activity for most people, and therefore a domain for which most people have strong intuitions for. Moreover, this domain has a near unique property that allows a user to model the behavior they wish to find. We found that, in many cases we could glean intuition as to how a driver's behavior would reveal itself in telemetry by simply "puppeteering" a smartphone equipped with an app that shows its acceleration and gyroscope readings. For example, by modeling a 3-point turn by sliding the smartphone on a desk, we can see that this behavior best revels itself on the surge axis as three consecutive bumps.

## 3.2 Occupational Health Sensing and Problems

Figures/arcplots.pdf

0.5 0.5

Figures/genomics.pdf

# 4

# Data Description and Management

Explain our sources of data to explore the methods developed in this work. Show why you chose this type of data and which purpose it has been used.

## 4.1 Public Datasets

### 4.1.1 Classification Benchmark - UCR

### 4.1.2 UCI Machine Learning Repository

### 4.1.3 Physionet

### 4.1.4 CPD Benchmark

## 4.2 Acquired Datasets

### 4.2.1 Office Job Dataset

### 4.2.2 Industrial Job Dataset

# 5

# Detection of Events and Summarization of Time Series

Have a small introduction about the problematic and inspiration in time series from the audio domain.

Será que poderia colocar isto sobre um tema maior? e depois ter tools desenvolvidas nesse ambito? Tools: Annotation, Segmentation, One-click segmentation of periodic events, summarization and profiling

## 5.1

### 5.1.1 Feature Representation

The extraction of relevant events from time series starts by computing the ??. This matrix has structural information, which provides pointers of the presence of the aforementioned events, detectable within the next steps. Figure **??** summarizes the steps involved in calculating the ??.

#### 5.1.1.1 Feature Extraction

As mentioned in section **??**, an event is defined as a significant change in the properties of the time series. Therefore, our proposed hypothesis relies on evaluating the changes that occur in a group of features of one or multidimensional time series of a record. The events' extraction depends on the richness of the set of features into translating the changes and disruptions of the signal. Behavioural changes might be related with different types of features (e.g. some changes might only be related with the mean, but others might be related with the frequency).

For example, the average feature translates changes of a group of samples through time, while the standard deviation the local disruptions. Conclusively, as a feature may be sensitive for a type of change, the type of features should be diverse to identify a multivariate set of events and scan different types of signals. For this purpose, we used

Figures/figure_ssm_scheme.pdf

, being $K$ the number of features used. The feature number is also associated with a shape (circle, triangle, etc...). The features can be extracted on multivariate records, being $M$ the number of records used. Each feature is positioned as a row on the $F_M$. Then follows the ?? computation.

Figure 5.1: Main process to reach the ??. The information needed to calculate the ?? are the record and the input parameters: the window size (w) and the overlapping percentage (o). The first stage involves the feature extraction process, based on $w$ and $o$ values. Features are extracted on each window (1, 2..., N), being N the total number of windows. From the first window ($w_1$), are extracted features (f1, f2..., $f_K$

the available features from the TSFEL [**barandas_tsfel_2020**] Python library presented in the Feature Table xxxx, in Annex 2.

The features are extracted in each sliding window with size $m$, with an overlap of size $o$ percent. These two parameters have a large influence on the results. The first defines the time scale at which features are extracted. Consequently, the larger the window's size, the larger the time scale at which feature values change. The second parameter defines the pixel-resolution of the resulting feature-signal, reducing the amount of information (down-sampling) with the decrease in the overlap value.

After all the features are extracted, the output will be a feature matrix ($F_M$), where the rows represent each feature and the columns the corresponding time-window. Features extracted from a multidimensional record are ordered in the $F_M$ as rows as well. The total number of rows can be, at maximum: $f \times k$, being $f$ the number of time series being analysed and $k$ the number of features extracted, as illustrated in Figure **??**. Finally, each row is z-normalized.

#### 5.1.1.2 The Self Similarity Matrix

After grouping all the features extracted, the next stage highlights the differences with the time series segments through the creation of the ??. This process consists in comparing each segment with the others within the record. Since each column of the $F_M$ is the feature characterization of each time-window, the comparison between segments is achieved by calculating the dot product between the transposed $F_M$ and itself:

$$SSM = F_M^T F_M \tag{5.1}$$

The dot product gives a similarity score based on the feature values of each time window. Cells of the ?? with higher similarity scores indicate that the corresponding time windows have similar feature values [**audiolabs1**, **audiolabs2**]. As a result, the ?? provides rich visual information, highlighting structures, such as blocks and diagonals, that describe the signal's morphological behavior over time.

In Figure **??**, the main structures are illustrated and highlighted in an example of an *??* [**audiolabs1**]. These structures are used to extract relevant information about the time series. Our proposed methods for annotation take advantage of these main structures to extract the desired information:

- **Sub-Diagonals -** When high similarity diagonals appear on the ??, parallel to the main diagonal, we can infer that the columns and corresponding rows of the ?? segmented by the *sub* diagonal have similar feature values. Therefore, the *sub* diagonals' presence is a means of detecting reoccurring patterns of the time series. In Figure **??**, the events **e1**, assigned as orange circles, correspond to events that represent a periodic pattern on the time series.

Figures/Example_SSM.png

.

Figure 5.2: Description of informative structures of the *SSM*. Based on the ?? presented in Figure **??**, we show a simplified view with highlights on the relevant structures. The record has 4 main structures: A - homogeneous segment, which corresponds to the BVP periodic signal; B - homogeneous segment of missed data; C - homogeneous segment with detachment of the sensor. The boxes (blue) highlight homogeneous blocks while the sub-diagonals (orange) highlight periodicity in the segment. *nf* and *sf* indicate that the novelty and similarity functions are computed based on this information. Segment C has a cross-pattern, which indicates periodicity and symmetry

- **Blocks -** Blocks are represented by square shaped structures on the ?? that highlight areas of the time series with a homogeneous morphology, i.e., a block has a homogeneous behavior on its length. The change between block structures along the main diagonal indicates a relevant change of morphology and behavior on the time series. We can detect events on the time series associated with transitions between homogeneous blocks of information. Additionally, blocks with a similar color are an indication that the corresponding segments on the time series have similar properties. In Figure **??**, the block structures along the diagonal that can be tagged as "A" and "B" respectively, represent different homogeneous areas of the time series, for instance, in block "A", the structure is periodic, as diagonals are present in it. Whereas in block "B", it is not periodic and represents an area of the signal different from block "A".

Several methods can be used to extract the valuable information provided by the ??. As aforementioned, (1) the information provided by the diagonals enhances the presence of periodical behavior, (2) the presence of blocks indicates an area of the time series with a homogeneous morphology, (3) transition between blocks indicate the presence of a significant change on the time series and (4) blocks and segments of the ?? that are similar have higher matching values, while dissimilar ones have lower matching values.

In this work, we applied several strategies on the ?? to extract this information and validated these approaches in multiple datasets.

### 5.1.2 ?? and event search

The ?? is a data representation of the record that allows extracting relevant information, such as the identification of relevant events. Several techniques can be used to extract most of this available information. These differ according to the structures we are searching for, namely, when searching for *off* diagonals on the ??, we are searching for changes representing periodic events, while when searching for blocks, we are searching for relevant changes on the time series or homogeneous areas. The first process requires the identification of the starting point of each diagonal, while the second process applies a strategy that involves sliding a square matrix, with a checkerboard pattern, along the main diagonal, to search for changes between homogeneous blocks. This reasoning has been inspired by several works applied in the context of musical structure analysis from audio signals, namely the works of *Meinard Muller et al.* [**Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**] and applied on an extended set of time series. The concepts presented for event search on time series are based on repetition search, homogeneous search and novelty search. Each of the used strategies is explained further.

Figures/KernelDescription.pdf

Figure 5.3: Description of the matrix (kernel) used to compute the *novelty function*. The checkerboard pattern is achieved by combining kernel $K_H$ - measure of homogeneity; and $K_C$ - measure of cross-similarity. The resulting kernel ($K_N$) is combined with a Gaussian function to generate $K_G$. The Figure is based on the works of *Mueller et al.* [**Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**]

#### 5.1.2.1 Change Point Event Search

The search of change point events is inspired by a method used in musical structure analysis and presented by *Foote et al.* [**foote2000**]. The process involves searching for transitions between block structures using a sliding square matrix. The result is a 1 dimensional function designated *novelty function*.

As aforementioned, block structures represent homogeneous segments of the time series, therefore by identifying the transition between blocks we are detecting moments in time with a relevant change on the time series properties.

As showed on Figure **??**, block transitions along the diagonal are represented by a checkerboard pattern. Detecting such patterns can be made by correlating a standard checkerboard matrix with the diagonal of the ??. For this, a sliding squared matrix, designated kernel, is used. As illustrated in Figure **??**, the kernel has a checkerboard pattern and is combined with a Gaussian function to add a smoothing factor. The kernel, $K_N$, is a combination of two different kernels: $K_H$ and $K_C$. The first is responsible for identifying the homogeneity of the ?? in each side of the center point along the diagonal. The higher the homogeneity, the higher will be the values in these sections. The latter measures the level of cross-similarity, returning higher values in cases of high cross-similarity. Therefore, when sliding the kernel $K_N$ along the diagonal, a higher correlation value will be returned when it reaches a segment of the ?? with a similar checkerboard pattern. The result is the mentioned *novelty function* ($nova$) [**Dannenberg2008**, **Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**].

As showed in Figure **??**, the kernel in position **A**, which is placed on an area of high homogeneity, returns a value close to 0 when summing the product between it and the section of the ?? it overlaps. In the other end, in position **B**, the kernel reaches a transition segment of the ??, which results in high correlation values. The *novelty function* is high in these transition segments [**Dannenberg2008**, **Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**].

Each section of the kernel has the same size, $L$, being the total kernel size configured by $D = 2 \times L + 1$, with $L \in \mathbb{N}$ . The kernel has an odd size to adapt zero values in centered points. It also has total size D × D, being $K_N$ defined by the following function [**Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**]:

$$K_N(i, j) = (a_i) \cdot (b_j) \tag{5.2}$$

, being $a, b \in [-L : L]$ and "" representing the sign function, which indicates the sign of the value (1, 0 or -1).

A radially symmetric Gaussian function is used to smooth the Kernel, with the following equation [**Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**]:

$$\phi(s, t) = \exp(-\frac{1}{2L\sigma^2}(s^2 + t^2)) \tag{5.3}$$

, being $\sigma$ the standard deviation, equal for both $x$ and $y$ dimensions of the matrix, $L$ the size of each kernel's section, and $s$ and $t$ the position in the $x$ and $y$ dimensions, respectively. The final kernel is computed by point-wise multiplication with the Gaussian function:

$$K_G = \phi \cdot K_N \tag{5.4}$$

After defining the kernel, it is used to compute the novelty function and reveal moments where relevant changes occur on the time series. The novelty function is calculated by correlating the kernel with the diagonal of the matrix:

Figures/KernelSlide.png

Figure 5.4: The process to compute the novelty function is described. Kernel $K_G$ is slided along the diagonal of the ?? to compute the *novelty function* presented as the bottom sub-plot. Positions A and B show the effect of block transitions on the *novelty function*. Figure based on the works of [**Dannenberg2008**, **Mueller15_FMP_SPRINGER**, **MuellerZ19_FMP_ISMIR**].

$$nova(n) = \sum_{i,j=0}^{2L+1} K_G(a_i, b_j) SSM(n + a_i, n + b_j) \tag{5.5}$$

, being the sample of the novelty function $n \in [0 - N]$ and $a, b \in [-L : L]$.

The change point events are represented by local maxima (peaks) in the novelty function. The location of each peak can be calculated using a peak finding algorithm. In our work, we used a naive approach that detects peaks greater than a defined threshold.

#### 5.1.2.2 Periodic Search

As aforementioned, *sub* diagonals indicate the presence of similarity and reoccurring patterns can be visualized on the ??. The moment in time the *sub* diagonals start indicates the position at which the period of the pattern begins. In order to find the periodicity available by the *sub* diagonals, we compute the similarity function, $s_f$, which is calculated by summing the values of the ?? column-wise, being each element of the $sf$ calculated by:

$$sf(x) = \sum_{i=0}^{N} SSM_{ix} \tag{5.6}$$

where $i$ is the column position for the sum, $sf_j$ the sample of the function at position $j$ and $N$ the size of one of the dimensions of the *??*. As segments with similar morphology will be similarly described by the extracted features, the columns will have a similar representation, hence a similar value on the *sf*. In cases where the time series is periodic, the similarity function will enhance this behaviour by having valleys at the moment the diagonal starts. The identification of events related with the periodicity of a time series is then possible by searching for valleys on the similarity function.

### 5.1.3 Validation Metrics

The validation metrics differ depending on the type of events we are searching. The ones used for evaluating the performance when using the *novelty function* are calculating the recall, precision and F1-measure in detecting the ground truth events, as well as calculating the distance at which the detected events are from the ground-truth events. In the other hand, the detection of periodic events by means of the *similarity function* is validated by calculating the number of correctly segmented periods. Each of this validation strategies are explained further.

#### 5.1.3.1 Validate Events Detection

Each record is compared sample by sample and the performance is evaluated by considering the number of true positives (TP), false positives and negatives (FP, FN). As the ground-truth event is one sample, we added an error margin to have a proper validation. This margin dictates if an event can be TP, FP or FN. In this work, we used the window size as the chosen margin. The estimated events are considered one of the following categories:

- TP - is counted when the estimated event is in the margin around the ground-truth event;
- FP - is counted whenever it is out of a margin around the ground-truth event, or when there is more than one estimated event inside the margin;
- FN - is counted when there is no estimated event inside the margin of the ground-truth events.

From the count of TP, FP and FN, we are able to calculate performance metrics, such as Precision, Recall and F1-measure, which are calculated as follows:

$$Pre = \frac{TP}{TP + FP} \tag{5.7}$$

$$Rec = \frac{TP}{TP + FN} \tag{5.8}$$

$$F1 = 2 \cdot \frac{Pre \cdot Rec}{Pre + Rec} \tag{5.9}$$

Additionally, the distance of the TP events from the ground-truth events is calculated with several distance-based metrics, namely the mean-absolute-error (MAE), the mean-squared-error (MSE) and the mean-signed-error (MsE):

$$MAE = \sum_{i=1}^{k} \frac{|g_i - e_i|}{k} \tag{5.10}$$

$$MSE = \frac{1}{k} \sum_{i=1}^{k} (g_i - e_i)^2 \tag{5.11}$$

$$ME = \frac{1}{k} \sum_{i=1}^{k} (g_i - e_i) \tag{5.12}$$

The precision measure is relevant to indicate if the method is able to only estimate events that belong to the ground-truth category, while the recall measure is an important indication of how many ground-truth events are missed in the estimation of the method. Both measures are combined in the F1-measure. The true negative samples are not considered since most of them would be correctly estimated, which would wrongly improve the accuracy.

The distance-based metrics evaluate how far are the TP from the corresponding ground-truth events (MAE and MSE) and which is the direction of estimation of events (if before or after the ground-truth events - ME).

These are the metrics employed for all datasets except for *Dataset 8*, for which the evaluation was made considering their internal measures, as explained in [**cpd_alan**].

### 5.1.4 Self-Similarity

### 5.1.5 Novelty Search

### 5.1.6 Periodic Search

### 5.1.7 Experimental Evaluation

In this section, we present several examples in how this method is useful for the segmentation of time series. The reader will appreciate that we also provide a measure of the algorithm's performance considering ground truth events (as presented in Section **??** and **??**), while also testing our proposed solution by comparing it with several methods for change point detection from the *Turing Change Point Detection Benchmark* [**cpd_alan**].

In addition, we give insights about how this method could be used to summarize a time series and assist the labelling process of time series.

We make available all the code and results on the online repository.

This section is divided into three main categories: (1) Validate the usage of the ?? on segmentation with several use-cases; (2) Provide intuition over the parameters used, explain specific use-cases results, show the difference when using multi-dimensional time series and comments on the scalability and speed (3) coments on how to explore the usage of the ?? for summarization and labelling.

### 5.1.8 ?? for Segmentation

As presented in Section **??**, the ?? highlights areas of information with similar behaviour as homogeneous blocks. The detection of the transition between these blocks of information is a direct application of this method to change point detection. This results in the extraction of relevant events that segment the time series in differentiated sections. In the other end, sub-diagonals indicate the presence of periodicity. Hereby we present several use-cases of segmentation with the proposed method. Overall results will be presented and discussed.

#### 5.1.8.1 Visual Examples in selected Use-cases

**Human Activity Segmentation**

The example presented in Figure **??** shows the usage of the ?? on a record of Dataset 2.

In this example, the method was applied to all the 3-axis of the accelerometer data. We only show the X-axis, which is described with the bottom sequence of activities as captioned in Figure **??**.

The ?? was computed using a time scale parameter of 250 samples, and an overlap of 95 %. Blue indicate segments with higher similarity. Along the diagonal, these blocks are visible and change point events are estimated as the transition between these, highlighted by the novelty function. The kernel used for this detection had a size of 45 samples.

Figures/HAR_example_2.png

Figure 5.5: Change point event detection strategy applied on the SSM to search for change point events. The sequence of activities is presented as follows: $Sitting \rightarrow Laying \rightarrow Walking \rightarrow Upstairs \rightarrow Downstairs \rightarrow Sitting \rightarrow Standing \rightarrow Laying \rightarrow Walking \rightarrow Upstairs$. The input variables used are $time_{scale}$=250 samples, $kernel_{size}$=45 samples, overlap=95%

In this example, we can identify that the detected change point events match with the activity transitions. Although all transitions are visible on the novelty function, the ones that correspond to transitions between similar segments of activities are harder to find, namely the transitions between walking activities. This is plausible since the properties of these segments are similar and the morphological difference is not as significant as when transiting between dissimilar activities (between *Laying* and *Walking* for example). There are ways of enhancing these differences, which will be explained further.

Any change with a significant change in properties will be detected by the proposed method. As presented in Figure **??**, at the end of the time series, the period in which the subject was performing the *Walking upstairs* activity is affected by other changes in the time series properties. These are significant and also correspond to block transitions, which are also evident in the novelty function. The proposed strategy, being unsupervised, is

Figures/BVP_example_2.png

Figure 5.6: ABP signal change point detection. The parameters used were a size of 5000 samples, with an overlap of 75% and a kernel size of 25 samples.

sensitive to any change, as long as it is observed as a significant change in the signal's properties. In this particular example, since the events do not correspond to transitions between the considered sequence of activities, they were considered as FP.

#### Arterial Blood Pressure

This is an experiment on arterial blood pressure data from a physionet dataset [**tilt**, **PhysioNet**]. There are change points indicated as a modification in the physiological signal due to a change in the posture of the subject. The change point is very well discovered with our proposed method, as signaled by the ground truth.

#### Noise Detection

In ECG signals it is common to find artifacts [**dataset6**]. In this case we segmented

Figures/ECG_example2.png

Figure 5.7: ECG signal from Dataset 7. The parameters used were a size of 500 samples, with an overlap of 95% and a kernel size of 50 samples and 10 samples, respectively.

noise from the ECG signal due to a jump performed by the subject with the novelty function. The segment is showed as a large area of red in the matrix, which indicates a dissimilarity with the rest of the signal, as well other structures from the ECG, which can be segmented with a smaller kernel size. In addition, we can compute the similarity function to segment the periods of the ECG.

#### Motion Analysis in the Industry

Another example is from an industrial scenario, where a worker performed cyclic activities and was interrupted several times [**antonio**, **santos_explaining_2020**]. The novelty function is able to segment the areas of work versus pause and the similarity function indicates where the working cycles occurred with local minima in the similarity function.

Figures/example_industrial.png

Figure 5.8: Example of a worker performing cyclic tasks in an industrial setting. The worker has interruption in his/her work which are annotated with the red labels. The novelty (nova) and similarity (sim) functions are computed from the ?? generated with a window size of 2500 samples, overlap of 85% and kernel size of 50 samples.

#### 5.1.8.2 Multidimensional Segmentation

The proposed method accepts both single and multidimensional records. The difference regards the number of features extracted. As presented on Figure **??**, the same set of features are extracted for each time series of the record and combined in the $F_M$.

Using a single time series of a multivariate record is optional and depends on the detection's purpose. In some cases, using a single time series from a multidimensional record can lead to missing relevant events undetected. An example of this can be seen on Figure **??** with record *"Occupancy"* from Dataset 8.

The record is a multi-dimensional time series that measures room occupancy based on temperature, humidity, light and CO2. All events can only be detected if using several time series of the record [**cpd_alan**]. On Figure **??**.left, a single time series was analyzed by the proposed method to detect relevant events, while Figure **??**.right presents the application

Figures/example_occupancy.png

Figure 5.9: Proposed method applied on "*Occupancy*" record of Dataset 7. A single time series of the record is used to extract events.

of the method to all the time series of the record. The results are different because the information from all the time series is combined, while with the single dimensional record, the detected events resulted from the information available on the single time series.

#### 5.1.8.3 Event Detection Performance

The proposed method has been tested on publicly available datasets from different domains to infer its performance on detecting change point events. These datasets have categorized labels that were used to generate ground-truth events. These include different contexts (HAR, Hand Posture, Noise Detection, etc...) and different types of data (Inertial data, EMG and ECG). More details are given on the problem associated with each dataset on

| Dataset | Signals | # Ch | Task | TP | FP | FN | Prec (%) | Rec (%) | F1 (%) |
|---|---|---|---|---|---|---|---|---|---|
| Dataset 1 | ACC | 3 | HACP | 98 | 16 | 16 | 0.860 | 0.860 | 0.860 |
| Dataset 2 | ACC-GYR | 6 | HACP | 157 | 18 | 22 | 0.897 | 0.877 | 0.887 |
| Dataset 3 | ACC-GYR | 6 | HACP | 1378 | 313 | 263 | 0.815 | 0.840 | 0.827 |
| Dataset 4 | ACC-GYR | 12 | HACP | 499 | 71 | 38 | 0.875 | 0.929 | 0.902 |
| Dataset 5 | EMG | 8 | Act/Rel | 309 | 0 | 72 | 1.000 | 0.811 | 0.811 |
| Dataset 6 | ECG | 1 | Noise | 132 | 25 | 10 | 0.841 | 0.930 | 0.883 |
| Dataset 7 | ECG | 4 | Noise | 21 | 2 | 3 | 0.913 | 0.875 | 0.894 |
| Total | N.A. | N.A. | N.A. | 2629 | 465 | 386 | 0.850 | 0.872 | 0.861 |

Table 5.1: Overall results for the performance of the method on change point detection. The dimension of the records is presented on the column # *Ch*, as well as the types of signals used and the task in which applied (HACP - Human Activity Change Point detection; Act/Rel - Activation/Relaxation of the EMG detection and Noise detection).

| Dataset | $T_s$ (s) | $MAE/T_s$ | $MsE/T_s$ |
|---|---|---|---|
| Dataset 1 | 5 | 0.53 | -0.12 |
| Dataset 2 | 10 | 0.29 | -0.07 |
| Dataset 3 | 1 | 0.34 | -0.04 |
| Dataset 4 | 25 | 0.23 | -0.00 |
| Dataset 5 | 1 | 1 | -0.13 |
| Dataset 6 | 10 | 0.12 | -0.09 |
| Dataset 7 | 1 | 0.17 | -0.06 |
| Average | N.A. | 0.32 | -0.07 |

Table 5.2: Distance error as a ratio of the time scale ($T_s$) for the detected TP.

Section **??**.

The method has been computed in the same conditions and by following the same procedure for all records of all datasets. The features used have been the same for each record, varying the time scale parameter, the overlap size of the sliding window and the kernel size parameter. The peak detection strategy was the same for all records, which is based on a threshold value. The threshold value varied for each record.

Results for publicly available datasets are presented in Tables **??** and **??**. Table **??** indicates the performance in detecting the change point events.

## 5.2 Time Series Profilling

### 5.2.1 Elements with Relevance

### 5.2.2 Minimalist Design

(Find better words to describe this)

### 5.2.3 Summarize Time Series

## 5.3 Further Developments

# 6

# Text Mining Time Series

## 6.1 Synthatic Search on Time Series

We have made 2 approaches, one which uses a translation of the time series into the symbolic domain and another in which we associate extracted features with words.

I AM NOT HAPPY WITH THIS SEPARATION...IT FEELS LIKE IT COULD HAVE A MORE HOMOGENEOUS STRUCTURE

A human language is foremost a means of communication, in which the information is represented by sentences, composed by words that can be broken into sequences of symbols. The diversity of possible arrangements of symbols and words gives the versatility in the process of transmitting information. The analysis of how symbols and words can be arranged in order to have a valid structure and comprise meaningful information involves the study of grammar and meaning davidCrystal.

Time series are, in their turns, carriers of information about a certain measure. These comprise sequences of ordered real domain numerical data observed during a given temporal interval, which are typically plotted as variations in amplitude. As aforementioned, the visual perception of the morphological behaviour of these series is, in many cases, enough to solve the problem and find the pattern that is being searched.

In terms of morphology, many attributes can be extracted from the visual perception of the signal, such as rising and falling slopes, concavity, direction, amplitude thresholding, frequency, time and amplitude range of a slope, among others. For instance, we can identify positive peaks by finding a rising slope followed by a falling slope, which is precisely the mechanism developed in Horowitz *et. al.* for peak detection in electrocardiography Horowitz.

The combination of these attributes into a sequence of primitives is a symbolic characterization of the signal that enables the use of text parsing tools, such as regular expressions, for searching the desired pattern.

### 6.1.1 Regular Expressions for Time Series

In 1943, McCulloch and Walter Pitts made the first theoretical description in a logic interpretation of the physiological events of neuron networks that served as inspiration for Kleen (1956) to create a set of rules that represent a finite state machine. Kleene described the nerve net as an arrangement of a finite number of neurons, where each has a sequence of states/events represented by integers. The state's values are influenced by the sensory response to the environment Kleene, and are said to be equally spaced in time. This algebraic description of neural nets can be extrapolated for time-series, in which the sequence of numbers is abstracted as a sequence of states to which values correspond to the sensory response of the environment.

The set of regular rules is a way of describing a specific sequence of states in the neural net - *a pattern*. This functionality has been extended into the field of text processing, in which this set of rules is able to describe a pattern as a sequence of characters, designated as a regular expression. Using a symbolic representation to characterize the sequence of states of time series in multiple attributes, regular expressions can be extended as a time series's parser to search patterns on it Thompson.

### 6.1.2 A Tool of Thought

In 1980, Kenneth E. Iverson has discussed the importance of notation, nomenclature and language as tools of thought APL1. A regular expression is a good example of a tool of thought, by expressing the recognition of a pattern into a sequence of characters, but other examples can be given, such as in chemistry, botany and especially in mathematics.

E. Iverson believed that, although mathematical notation is not universal and unambiguous, it provides one of the best-known and best-developed examples of language as a tool of thought. With this in mind, he developed a programming language called APL (A Programming Language), which has the advantages of being universal and unambiguous, and incorporated the principles of mathematical notation.[**referencia**]

One of the fundamental characteristics of this tool is the provision of graphic symbols for the execution of functions and operations, which are meant to express the thought of the user in solving a problem. The tool presented in this work is inspired by this reasoning and uses graphical symbols in the pre-processing and symbolic connotation steps. With this, the proposed tool profits of E. Iverson ideas to be intuitive, simple to use yet complex enough to reach the desired end, being a powerful tool of thought for query search in time series.

### 6.1.3 The Syntactic Pattern Quest

In this study, we propose a tool that focuses in ease simple query search tasks in time series, which we refer as ??. This is achieved by an innovative methodology, where the user gives a syntactic nature to time series, which turns the search procedure less verbose and more related with the reasoning of the user in recognizing the desired pattern.

### 6.1.4 Time Series Representation

## 6.2 Towards Natural Language for Pattern Search

## 6.3 Classification of Time Series Documents

# 7

# Examples of Ergonomy

1 - Example of the data we acquired in Volkswagen when we arrived and tried to compare two workstations. We were trying to find specific cyclic moments and labelled it by hand. We can use this tool to make this identification (SSM):

2 - Describe the pattern by means of words or a regular expression

# A

# *NOVAthesis* covers showcase

This Appendix shows examples of covers for some of the supported Schools. When the Schools have very similar covers (e.g., all the schools from Universidade do Minho), just one cover is shown. If the covers for MSc dissertations and PhD thesis are considerable different (e.g., for FCT-NOVA and UMinho), then both are shown.

# B

# Appendix 2 Lorem Ipsum

This is a test with citing something [**ecoop12-dias**] in the appendix.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

I

# Annex 1 Lorem Ipsum

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

year Grammar of Time Time Series Data Mining in Steroids John Doe