Rui Pedro Sousa Varandas

Bachelor Degree in Biomedical Engineering Sciences

Evaluation of spatial-temporal anomalies in the analysis of human movement

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

Master of Science in
Biomedical Engineering

Adviser: Prof. Dr. Hugo Filipe Silveira Gamboa, Professor Auxiliar, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

October, 2018

Evaluation of spatial-temporal anomalies in the analysis of human movement

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

With the conclusion of this phase of my life, I would like to acknowledge every person who made it possible.

Firstly, I would like to thank my advisor, Professor Hugo Gamboa of FCT-NOVA, for the guidance, support and exigence during the course of this project. I thank especially for all help provided in times of doubt, for guiding my focus in the right direction and for all the suggestions and opinions given throughout this time. I also thank the chance to learn more about machine learning and signal processing techniques and the inspiration to constantly research and learn more about these topics.

I would also like to thank Fraunhofer Portugal for giving me the opportunity to work in its offices and providing me with all the tools necessary to develop this research. The working environment was always helpful and the chance to contact with a research environment in such an early stage of my life was excellent for acquiring knowledge about working practices and the challenges and rewards of research work. I am truly grateful to everyone at Fraunhofer Portugal for being so accessible in times of need and especially to Duarte Folgado, without whom this thesis would not be possible. I appreciate all the patience, recommendations and, specially, minutia analysis of every question I made, complex or not. I would also like to thank Marília Barandas for always facilitating and helping with everything was necessary throughout this time. Furthermore, I thank the students team that accompanied me in this laborious task for making it easier, for all help and for the laughs. You were truly outstanding!

To my friends, I thank the hours spent studying, for new insights and motivation, for the time spent playing table football and for all the jokes. Thank you all for the support and help during the last years.

I would like to express my gratitude towards my grandparents who always encouraged me to keep going in order to be a better person. For all the stories and smiles, thank you!

Lastly, I thank my family, especially my parents, my brother and sister, for all the motivation and support, and for the high expectations deposited in me, which made me want to always be better and always do more!

Abstract

The dissemination of Internet of Things solutions, such as smartphones, lead to the appearance of devices that allow to monitor the activities of their users. In manufacture, the performed tasks consist on sets of predetermined movements that are exhaustively repeated, forming a repetitive behaviour. Additionally, there are planned and unplanned events on manufacturing production lines which cause the repetitive behaviour to stop. The execution of improper movements and the existence of events that might prejudice the productive system are regarded as *anomalies*.

In this work, it was investigated the feasibility of the evaluation of spatial-temporal anomaly detection in the analysis of human movement. It is proposed a framework capable of detecting anomalies in generic repetitive time series, thus being adequate to handle Human motion from industrial scenarios. The proposed framework consists of (1) a new unsupervised segmentation algorithm; (2) feature extraction, selection and dimensionality reduction; (3) unsupervised classification based on DBSCAN used to distinguish normal and anomalous instances.

The proposed solution was applied in four different datasets. Two of those datasets were synthetic and two were composed of real-world data, namely, electrocardiography data and human movement in manufacture. The yielded results demonstrated not only that anomaly detection in human motion is possible, but that the developed framework is generic and, with examples, it was shown that it may be applied in general repetitive time series with little adaptation effort for different domains.

The results showed that the proposed framework has the potential to be applied in manufacturing production lines to monitor the employees movements, acting as a tool to detect both planned and unplanned events, and ultimately reduce the risk of appearance of musculoskeletal disorders in industrial settings in long-term.

Keywords: Time Series; Anomaly Detection; Human Motion; Unsupervised Learning; Manufacture.

Resumo

A disseminação de soluções relacionadas com a Internet das Coisas, como *smartphones*, levou ao aparecimento de soluções capazes de monitorizar os seus utilizadores. Na manufatura, as tarefas executadas pelos trabalhadores são compostas por conjuntos de movimentos predeterminados, formando comportamentos repetitivos. Adicionalmente, existem eventos planeados e não-planeados que causam a paragem desse comportamento repetitivo. A ocorrência de movimentos mal executados, bem como de eventos que possam prejudicar o sistema de produção são denominados como *anomalias*.

Neste trabalho, foi investigada a viabilidade da deteção de anomalias espácio-temporais na análise do movimento humano. Para isso, é proposta uma nova ferramenta capaz de detetar anomalias em séries temporais repetitivas no geral. Essa ferramenta consiste (1) num novo algoritmo não-supervisionado para segmentação de séries temporais; (2) extração e seleção de características e redução de dimensão; (3) classificação não-supervisionada baseada no algoritmo DBSCAN utilizado para distinguir instâncias normais e anómalas.

A solução proposta foi depois testada com quatro grupos de dados diferentes. Dois desses grupos são constituídos por dados sintéticos, enquanto que os outros são constituídos por dados reais, nomeadamente, dados de eletrocardiografia e movimento humano na manufatura. Os resultados obtidos mostraram que, para além de ser possível detetar anomalias no movimento humano, a solução proposta é genérica e, com exemplos, foi mostrado que pode ser aplicada em séries temporais repetitivas com pouco esforço de adaptação para diferentes domínios.

Os resultados obtidos mostraram que a ferramenta desenvolvida tem o potencial de ser aplicada em linhas de produção na manufatura para monitorizar os movimentos realizados pelos trabalhadores, atuando como uma ferramenta capaz de detetar tanto eventos planeados como não-planeados, conduzindo à redução de incidência de doenças musculosqueléticas em contexto de indústria a longo-termo.

Palavras-chave: Séries Temporais; Deteção de Anomalias; Movimento Humano; Aprendizagem não-supervisionada; Manufatura.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

The evolution of technology lead to the dissemination of smart systems that are widely used on an every day basis, such as smartphones or smartwatches, which have the ability to help users with their daily tasks. Most of these systems have the ability to collect data from various sensors, that might be used for monitoring the daily activities of users. The gathered data may be respective to the movement of the users, such as walking or running, and the environment around them, for example, light and atmospheric pressure. The integration of all information may be indicative of their daily routine.

Thus, the collected data may be periodic, due to routines, and any deviation from the expected behaviour is called *anomaly*. That behaviour may be the posture of its user, for example, and the monitoring allows to make ergonomic risk assessment. Therefore, the detection of this phenomena is important, because it is usually associated with defective practices and, hence, can either help to prevent it in the future or correct and take advantage of it upon detection.

The existence of anomalies is not restricted to acquired data originated by wearable devices. Typical examples include anomalies detected in electrocardiography (ECG) signals, which might indicate the presence of arrhythmias and, consequently, heart diseases, or anomalies in the traffic pattern of a computer network, which could mean that a virus is stealing information or using the computer resources to some end. In addition, these techniques can be used in military surveillance, credit card fraud or flawed motors [1], and have been widely studied for price manipulation in stock markets [2].

1.2 Motivation

In Portugal, 83% of workers are subjected to repetitive work of hand or upper limbs, being the principal physical risk factor in workplace, followed by standing positions for long periods of time and being subject of tiring and awkward position, corresponding to 71% and 47%, respectively [3]. According to the same study, the more transverse sectors to the principal risk factors are the construction work sector and manufacture.

In manufacture environments, there are sets of movements that integrate each task that are designed to optimise both ergonomics and production, with the objective of averting the appearance of musculoskeletal disorders (MSDs), which may affect muscles, bones and joints. Thus, once these tasks are exhaustively repeated in a typical work day, a (quasi-)repetitive pattern is formed. If this pattern is broken, it might indicate planned events, such as breaks or shift changes, as well as unplanned events, namely, wrongly executed movements or machine idle.

Work study is a research field, which is divided in two separate areas: method study and work measurement. Work measurement is concerned with the study of the amount of time needed to perform each task, while method study is concerned with the study of the methodology involved with each movement, improving ergonomics and efficiency. Both of these methods cope with the fact that, typically, they require the presence of an evaluator for the assessment of work cycles, which can affect the final report of the performed study.

Thus, an automatic system capable of identifying those wrongly executed movements would allow for the evaluator to study the periods in which there are suspicions of those events, bringing more objectivity and allowing a more thorough study of each anomaly, ultimately reducing the appearance of musculoskeletal diseases.

1.3 Literature Review

Anomaly detection on time series is a process that can be fragmented in multiple stages. These stages constitute the required steps necessary to most traditional anomaly detection algorithms. After data acquisition, it is usually performed segmentation which allows a more thorough analysis of the gathered data. Then, each segment is represented by specific characteristics that can characterise the nature of each segment. Lastly, these characteristics are presented as input to the anomaly detection algorithm which computes the anomaly score or the direct classification for each cycle.

1.3.1 Segmentation and Representation of Time Series

Segmentation is performed in order to capture local properties of the analysed signal, allowing a localised analysis. In some applications, segmentation is performed with the intent to reduce dimensionality, in which case, each segment is represented by a relevant characteristic. The most simple form of segmentation is the fixed window segmentation,

which, as the name suggests, is based on the segmentation of the time series in segments with similar length. This method requires only the selection of the window length. Then, if each segment is represented by its mean value, the representation is named *Piecewise Aggregate Approximation* (PAA) (example in Figure 1.1).

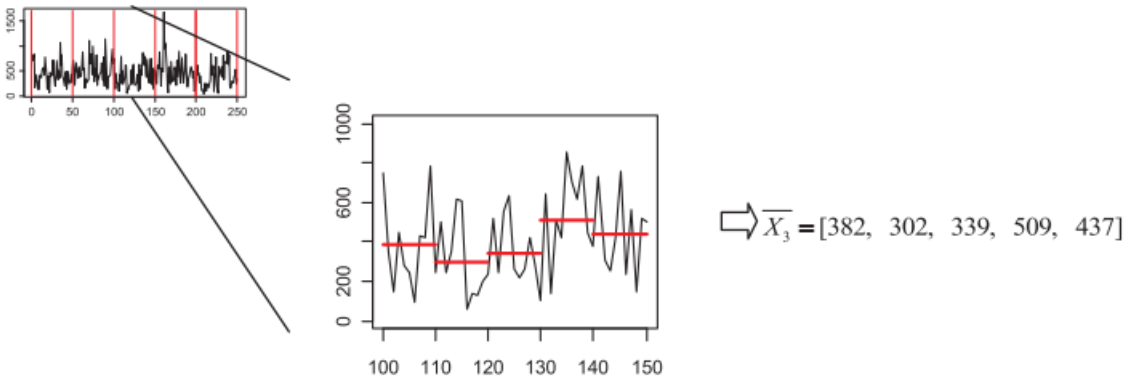

Figure 1.1: Visual representation of the PAA method [4].

Another widely used representation based on the same segmentation basis is *Piecewise Linear Representation* (PLR), which represents each segment of the time series by its linear regression or linear interpolation [5], which reduces information loss at the expenses of a higher computational cost. In [4] a new representation is proposed based on the same concept of segmentation, where each segment is re-segmented, this time in the amplitude domain. The segmentation of the amplitude domain is not performed in equal intervals, instead it is assumed that each segment follows a normal distribution and each amplitude segment has a similar number of points by using the concept of percentile to divide the amplitude domain data in the required number of sub-segments with equiprobability. Then, each sub-segment is represented by its number of points, the mean value and the variance. Each temporal segment is then represented by a matrix where each line corresponds to the amplitude domain segmentation and each column is the statistical feature of the considered segment, respectively. This technique is called *Piecewise Aggregate Pattern Representation* (PAPR) and it was developed on the scope of anomaly detection (see Section 1.3.2).

Symbolic Aggregate approXimation (SAX) is a technique used to discretise time series after segmentation [6]. Firstly, it is performed the PAA transformation and then, similar to PAPR, each temporal segment is segmented with equiprobability. The difference is that each segment is associated to a symbol, typically a letter, depending on the amplitude domain interval to which its mean value belongs, as exemplified in Figure 1.2.

An alternative is to segment the time series in segments that may have different lengths. *Adaptive Piecewise Constant Approximation* (APCA) is a techniques of this type,

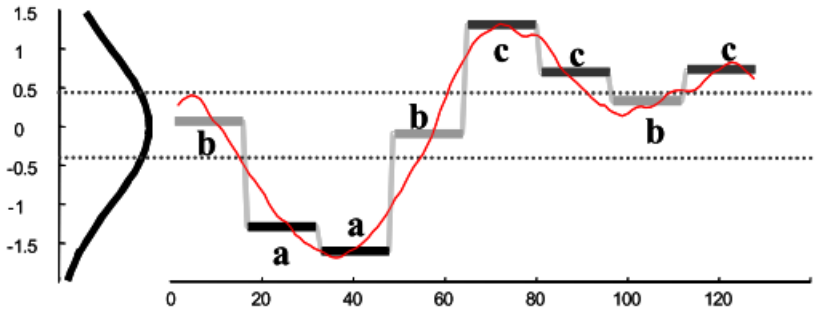

Figure 1.2: Visual representation of the SAX method [7].

which consists of finding split points depending on the approach to data and then, similar to PAA, representing each segment by its mean value. Similarly to PLR, this method allows to reduce the loss of information in comparison with PAA at the expenses of a higher computational cost.

These methods may use different approaches, such as sliding window, in which the first point of a window of the time series is anchored while the last point increases in time, being the window size increased. Then, given a model of the segment being searched, there is a stopping criteria which is usually the approximation error in relation to the model. Considering that the error increases with the growth of the window, if the error surpasses the defined threshold in i, the found segment consists on the time series from the anchored point to $i-1$. The process is then repeated anchoring the next window in i. This technique can also consist on the sliding of the first and last points of the window, while maintaining the window width, being the search performed in equal length segments. There are two other approaches: bottom-up and top-down. The distinction between sliding window and the other two processes is the fact that the sliding window may be used in online applications, while top-down and bottom-up can only be used in offline applications or batches. This is because, unlike the sliding window approach, the other two use the whole time series as starting point for the segmentation process. Bottom-up processes start by partitioning the whole time series in the finest part possible (usually in $N/2$ segments, being N the number of points in the time series) and then join adjacent segments. Analogously, top-down processes start by dividing the time series in parts and then, each of those parts are splitted recursively at the best location. Both top-down and bottom-up processes have the same stopping criteria as the sliding window approach [5].

Moreover, segmentation techniques can be classified as dictionary-based or dictionary-free. In dictionary-based methods, it is constructed a database of known sequences and the task is to find sequences in the time series that are identical to the ones present in the database. The methods described before belong to this category, but are focused on

the representation of time series. There are methods in this category that are exclusively focused on the segmentation process, such as Important Extreme Points, Perceptually Important Points and Polynomial Least Square Approximation. Dictionary-free methods do not require the construction of a database and the task is to find both the shape of the pattern and the segmentation points directly from the time-series. Examples of these methods are: Simple Kalman filters, Factorial Hidden Markov Models and inference based on a probabilistic representation. These methods tend to have high computational cost which prevents their utilisation in big data applications [8].

In [8], the authors developed a new algorithm to overcome the problem of high computational cost of typical dictionary-free segmentation techniques. For that it was explored the covariance structure of the time series data, Bayesian change point detection and temporal correlation on two different datasets. The key aspect that allows this algorithm on Big Data application is the use of Principal Component Analysis as the first step for dimensionality reduction. On the first dataset, which was an artificial dataset built by the authors, it was tested the effect of various parameters on the accuracy and performance of the proposed algorithm, where it was concluded that the accuracy is approximately $86,5\%$ without decreasing the performance for signals with the increase of the number of cycles or noise. Furthermore, with the second dataset, which is composed of several signals with 18 dimensions each corresponding to everyday activities, such as opening doors, using keys and eating, it was proved that the algorithm works and it was possible to differentiate distinct activities.

The authors of [9] developed a new unsupervised method to find patterns to segment human motion data acquired using a set of 10 inertial sensors placed in different joints, allowing a finer monitoring. It is assumed that the activities measured are repetitive, but once it is unsupervised, there is no *a priori* knowledge about the duration of each cycle. To estimate the window length of the underlying patterns, it is used the combined power spectral density to extract the dominant frequency. Then, a model is estimated based on the window with highest density to its k nearest neighbours, with Dynamic Time Warping as the distance function, and the process turns into a dictionary based approach in relation to the extracted model.

In [5] it was developed a new segmentation technique based on both Sliding Window and Bottom-up techniques, with the intent of getting the best of both approaches, which is the online application of Sliding Window and the high accuracy achieved by Bottom-up, that is called SWAB (Sliding Window and Bottom-up). This algorithm works in buffers, that are composed of data collected in an online fashion. The buffer size must be set to 5 or 6 times larger than the width of the segment that is being searched. Then, within the buffer the behaviour of this technique is similar to bottom-up. The sliding window effect is present in the sense that when a segment is found in the buffer, those

data points are discarded by the buffer and new data is included in it. The results show that the accuracy of this new technique is similar to the Bottom-up method.

In [10] a method was developed that enabled the assessment of the quality of segmentation techniques. The work provides a study about Important Extreme Points [11], Perceptually Important Points [12] and Polynomial Least Square Approximation [13]. In this work, it was added a new quality assessment parameter, the Percentage of Average Length Segments, to the eigth criteria already proposed in [14], based on classification metrics (accuracy, F1 score, Mathews correlation coefficient, precision and recall) and segmentation zone criteria: average segmentation count - determination of how many times the considered method segments inside a segmentation zone in average -, absolute segmentation distance - accumulation of the distance between segmentation points and segmentation targets - and average direction tendency - quotient of early and late segmentation points in relation to target points. The use of the new parameter, derives from the assumption that all segments have similar length, thus, it measures the percentage of segments which length falls in the range $]0,9L;1,1L[$, where L is the average value of the lengths of all cycles. The conclusion drawn by this work is that the new parameter can help to assess the quality of the segmentation because its results follow the same trend as the parameters proposed in [14]. Furthermore, it is concluded that there is not a best segmentation algorithm in general, and that it depends on the considered signal.

1.3.2 Anomaly Detection on Time Series

Anomaly detection methods may use specialised techniques or more general approaches, such as machine and deep learning algorithms. Specialised methods do not require explicit learning and utilise the concept of anomaly score, which is the anomaly degree of the analysed pattern, segment or time series, to indicate the presence of anomalies. However, in machine and deep learning techniques it is apprehended the normal or expected behaviour of a given process, and every segment or point that differs from the normal pattern is regarded as an anomaly. Typically, the difference between machine and deep learning is that the input for machine learning algorithms must be representative, meaning that the appropriate characteristics must be chosen prior to the training and testing phases, while deep learning techniques can extract those relevant features on their own or even apprehend the normal behaviour from raw time series.

There is extensive work in this field of study, and various surveys were already published [1, 15]. There are many application domains of anomaly detection such as in ECG signals and MRI images, where anomaly detection may have a direct effect in people's lives. In Chapter 2 a more thorough explanation of all concepts involving anomaly detection is presented.

HOT SAX is a method based on the SAX representation, that was developed in [10], to find discords which are sequences that are the most dissimilar to its k nearest neighbours. To construct the SAX representation it is used a sliding window with no anchored point and then the search is made on the transformation space, where each sequence is represented by letters. Thus, it is required a parameter d which indicates the number of discords that is expected to be found. Therefore, this algorithm, is able to find anomalies in time series, however it is necessary to know the number of anomalies to be found and, thus, to define the parameter *d a priori*. The main achievement of the research was to reduce the computational time required to find the discords, which was tested in 82 datasets of UCR Time Series Data Mining Archive.

The authors of [7] proposed a bitmap representation derived from the SAX representation of time series subsegments. These subsegments are constructed using a sliding window and then converting each to the discrete representation by letters. Then, the representation of each subsegment is transformed into a bitmap by a technique called chaos game representation. The anomaly score is given by the distance between two bitmaps, in which one of them can be known to be normal, being a supervised approach, or assuming that one of the bitmaps is normal, in which case it is an unsupervised approach. It is demonstrated that this method is promising by showing the results for a specific ECG signal.

In [4], it was developed the PAPR representation method with the intent to search for anomalous patterns in time series. It is assumed that normal segments have similar matrix representations of the statistical features mentioned in Section 1.3.1 and so, a similarity matrix is computed. Then, with the matrix, it is generated a random walk model that generates a graph, in which the nodes with most edges are the ones with more similarities, and an anomaly score is assigned to each pattern. The proposed algorithm was tested in 14 different real world datasets and compared with the PAA method, achieving higher results. While PAA method detected 15 anomalies, PAPR associated with Random Walk (PAPR-RW) algorithm was able to find 25 anomalies out of 27, and so, the sensitivity is approximately 92%.

In [16] an approach is proposed in order to detect anomalies in network source data. The objective is not only to find known attacks, which already have sets of rules that when applied can detect them, but also to detect unknown and previously unseen attacks in an unsupervised manner. For this, it was adapted the *Density-Base Spatial Clustering Algorithm of applications with Noise* (DBSCAN) clustering algorithm. With the developed algorithm it was possible to achieve a True Detection Rate of 93,7% and a False Detection Rate of 2,41%. However, it is not clear how the classification is performed, once the algorithm groups data points but does not classify them and, thus, the approach is not reproducible.

In [17] it was developed a solution to detect anomalies in human gait signals that could be used in an ubiquitous way. The process was to acquire data from an accelerometer placed on the person's waist, and then a computationally inexpensive classification system based on the k-nearest neighbour estimator was applied to make the classification of the gait pattern. The segmentation is based on step detection and, therefore, is domain specific. The validation of the algorithm was made in signals of 30 volunteers that made acquisitions in a controlled environment and 4 volunteers that made it in an uncontrolled environment during 48 hours. The results showed that the accuracy achieved in the controlled environment was better, as expected, and rounded 84% in the signals with anomalies.

Regarding applications with electrophysiology data, there are innumerable reports of arrhythmia detection in ECG signals [18], which can be considered anomalies. For example in [19], the use of an artificial neural network (ANN) with back-propagation was used in a supervised fashion in order to give an accurate classification of each heartbeat. The MIT BIH arrhythmia database was used and each heartbeat was extracted by selecting the 300 points around the R peak annotation provided by the experts that annotated the whole database, including the nature of each heartbeat, which served as input for the neural network. The achieved accuracy was approximately 96, 2% which was better than achieved in the compared works.

An extensive review of Deep Learning techniques for feature extraction and selection is present in [20], in which there is an extensive list of applications, including physiological signals, such as electroencephalography (EEG), magnetoencephalography (MEG), electrocardiography (ECG), and wearable sensors for health monitoring. The power of these techniques is well expressed in [21], where an *off the shelf* recurrent neural network was trained using various datasets from UCR TSC Archive composed of signals from different domains in order to capture relevant features in generic time series, constructing a domain-free tool, even though it is necessary to apply a classifier on the output of the network in order to obtain a classification.

In [22] it is proposed a method based on Long-Short Term Memory (LSTM) neural networks for the task of unsupervised anomaly detection on time series. In this work the LSTM was trained to predict the next values of a time series based on previous data and an anomaly score is attributed based on the difference between the expected result and the true time series. The only parameter that controls the outcome of the classification is a threshold for the anomaly score, which separates the considered normal behaviour from anomalous. The proposed approach was tested in three different datasets composed of a machine temperature data set, ECG signals and power demand readings, with promising results, although it is highly dependent on the threshold selection. Furthermore, deep

learning methods require large datasets in order to learn the expected behaviour and forecast time series.

1.3.3 Application in Industrial Scenarios

Industry has been suffering a transformation in recent years with the application of Information Technologies (IT) in association with Operational Technologies (OT) in order to bring higher production levels and facilitating work for employees while generating more income. One of the areas that could benefit with this growth is work study, which focuses on the study of the production process of employees with the intent to optimise it [23]. Work study comprises two main disciplines: work measurement and method study.

Method study is the area concerned with analysing the work, aiming to improve its effectiveness and efficiency [23]. The aim is to reduce superfluous movements that may cause loss of production time, reduce the human fatigue by reducing the effort needed to perform each task, increase performance of both human-labour and machine and improve the working conditions for employees.

Work measurement is the area concerned with the standard-time of execution of each task [23], which is the time required for a qualified employee to carry out a specific task at a normal pace, following a specific methodology and taking into account the fatigue, personal aspects and normal delays. There are three ways to perform the study: Work Sampling, Time Study and Predetermined Time Standards. Work Sampling is based on a statistical analysis of the work, in which random samples are analysed, representing the whole task, without the need of continuous analysis and observation of work. Time Study is the study of the time needed to carry out a specific task or movement under certain circumstances, allowing to obtain the standard time for each set of movements and tasks. Lastly, the Predetermined Time Standards technique requires the comparison of the time needed for each employee to perform each movement with synthetic libraries that contain extensive and detailed catalogues of movements and the times needed to carry them out. This enables the calculus of the exact human-labour required to do a specific job and to calculate the predicted income produced by those employees following the time-standards.

Both method study and work measurement have barriers in order to obtain accurate results due to social and technical factors [24]:

Social factors - *hierarchy of organisation culture*: decisions made by superiors are not always accepted and can lead to processes performed wrongly; *lack of full support*: work measurement depends on the commitment of both workers, superiors and technicians to be well done; *reluctance to measure*: sometimes employees might not understand the need for changing the production process because it has always been employed; *fear of job loss*: employees under test tend to be nervous because they see it as a way of finding inefficient workers rather than as a way of improving processes.

Technical factors - *tedium of measurement process*: sometimes the work supervisor has to supervise work during various hours and, if the job is repetitive and has many tasks along the day, the supervisor has to watch the same work being done for days, which can lead to a decrease in attention to details; *variation of work methods*: there are jobs that can be performed effectively utilising various methods, making it difficult to standardise a method to be measured; *ambiguity of process elements*: complex jobs require a great number of independent processes and finding these is not always easy. This can lead to a great variance number of processes in one job if evaluated by different supervisors. A correct analysis of the job is important for understanding which process can be improved rather than just considering that the whole job has to be performed better; *Shortage of needed samples*: the more the samples, more representative is the statistical analysis performed, so it is important to have a high number of measurements, although this is not always possible because the tests have to be done on multiple workers, shortening the number of samples from each one.

The number of barriers could decrease with an automatic system that could reduce the presence of an evaluator in the workplace, due to the overcoming of the technical factors of tedium of the measurement process and ambiguity of process elements, increasing the accuracy of work study.

With this intent there are several commercially available solutions. For example, Heptasense [25], which is currently in an experimental phase, uses video from surveillance cameras allied with artificial intelligence to perform the work study of the employees, meaning that it is required a high initial investment to acquire the sufficient number of cameras to monitor each workstation and each employee. Furthermore, it can have flaws due to the occlusion of cameras.

TotalControlPro™ is an example of a commercialised system that, with resource to a system of code bars that must be read each time the employee performs a specific task, is able to asses the timing of the performed work [26]. The greatest disadvantages of a system of this type is the initial investment required to implement the system in a factory and the time required to train the operators, due to the fact that it is not natural to work with and it introduces an additional movement on the work method.

In [27] a system based on video analysis is tested in which 5 cameras are strategically placed to acquire the images in a controlled environment. The analysis of these images allows to monitor the work and movements performed by the employee. The results were promising, but the tests were only conducted in a controlled environment without real manufacturing data. Furthermore, a visualisation dashboard was developed that allows an easy-to-read report of the tasks of the monitored operators.

Amazon Technologies, Inc. patented a new system that is able to perform this monitoring with resource to a system composed by ultrasound emitting bracelets and receivers placed in storage bins [28]. This system allows not only the monitoring of operators work, but also helps to direct them to specific bins that are being searched.

In manufacture there are few solutions for human motion monitoring using inertial measurement units (IMU). In [29] it was used a system composed of 7 IMUs, 2 electrogoniometers and 2 camcorders to assess the ergonomic conditions of 5 different workers for a given workstation. Even though there is a low number of samples considered, it was concluded that the ergonomic position and manner of work of those employees had to be changed in order to prevent musculoskeletal diseases, evidencing the need for a tool to detect these hazardous postures.

In this context, the work conducted in [30] established an innovative ubiquitous method to monitor human motion in manufacturing environments. An extensive analysis about the magnetic field, assessed from the magnetometer present in IMUs, revealed that its properties may be used to monitor the subject under analysis in a thorough manner when the magnetic field suffers relevant fluctuations in its workstation. Furthermore, the application of new distance based metrics for the analysis of the monitored repetitive signals allowed to extract cyclic events in the considered domains of application, using a dictionary-based technique. The extracted cycles may then be analysed in order to assess if it is normal or anomalous.

Furthermore, there are various solutions concerned with the detection and monitoring of anomalies in machinery originated data, that can indicate future failures that, when prevented can reduce in large scale the reparation costs [31] and it has been applied in quality control scenarios [32].

1.4 Objectives

Given the existing solutions, it is proposed with this project the monitoring of repetitive tasks. To analyse repetitive tasks, it is proposed the development of an unsupervised algorithm that is able to detect anomalies in generic periodic time series. Concretely, the objectives are: (1) explore the viability of unsupervised anomaly detection on human motion data acquired with resource to inertial sensors; (2) develop a new unsupervised anomaly detection algorithm on repetitive time series and provide a full analysis of the algorithm and its application in various data sets to test its generality; (3) develop a new unsupervised segmentation algorithm to extract cyclic events in repetitive signals.

While a generic tool may have greater difficulty to apprehend the normal behaviour of a given system, the fact that each workstation in a manufacture environment has different

sets of tasks and movements, makes it impracticable to develop specific algorithms for each workstation. Moreover, the unsupervised approach is used to let the system learn the behaviour of each employee without the necessity of prior knowledge or any labelling effort of an evaluator.

With this solution, it is intended to have a new work study technique that may also be used in various domains, such as human motion, ECG signals, predictive maintenance, fall detection in gait signals and behavioural changes. All this examples have in common the fact that all of them generate quasi-periodic signals in normal circumstances, if correctly measured.

There is a schematic in Figure 1.3 in which there is a brief overview of the techniques required for the proposed solution, the assumption on what data and level of knowledge about that data in which it is supposed to function and various examples of application.

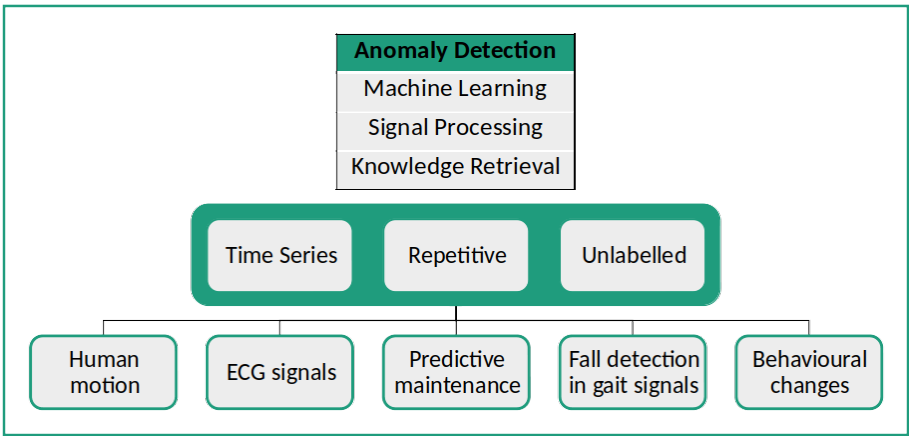

Figure 1.3: Overview of the proposed solution.

This algorithm will then be applied in industrial scenarios to help the prevention of appearance of musculoskeletal diseases and also to assess the training of new employees by giving statistics of their work.

1.5 Summary

As reviewed in Section 1.3.2, anomaly detection techniques rely on methods that either involve a high amount of parameters or the data may be so reduced that it leads to a significant loss of information, especially the methods involving data reduction techniques. The methods that showed to be more promising were the ones involving deep neural networks in the sense that the number of parameters is very low and the methods does not involve loss of information. Moreover, they could be applied to a great extent of data sets without modification, dispensing an expert to analyse a new given data set. However, the computational power of the first methods is lower than that of the deep learning methods, and the required training data required to train Deep Neural Networks is usually too large for new application domains.

Therefore, this dissertation will focus on developing a new algorithm capable of detecting anomalies in general data sets combining the aforementioned methods of preprocessing and classification. These methods will then be used in a specific manufacturing application in order to be validated in a real world industrial scenario.

1.6 Dissertation Outline

This dissertation is structured as shown in Figure 1.4. The structure is divided in 3 areas: basis and concepts, methods and outcomes.

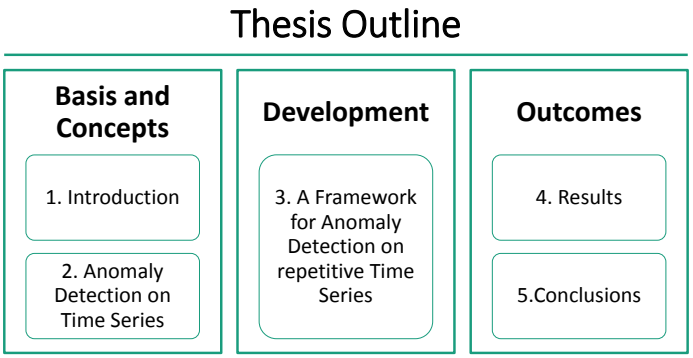

Figure 1.4: Dissertation outline.

The rest of this work is organised as follows: in Chapter 2 an extensive list of theoretical concepts is given with the aim to justify the choosing of the followed approach. In Chapter 3, it is described the developed framework for anomaly detection, and the explanation of each step. In Chapter 4, the obtained results are presented, as well as the description of the four data sets used to validate the developed framework. The conclusions of this dissertation project are outlined in Chapter 5 along with future work that is yet to be developed.

Chapter 2

Anomaly Detection on Time Series

In this chapter, it is intended to provide a better understanding of the concepts that support anomaly detection on time series and also the most widely used methods.

2.1 Time Series

Time series are sets of data points ordered in time. Formally, a time series composed by N data points is represented by

$$X := \{x_1, x_2, ..., x_N\} \tag{2.1}$$

where each x_i, $i \in \{1, 2, ..., N\}$ corresponds to a data point. These signals let us visualise the behaviour of the underlying process over time. Time series are, thus, important tools to understand real world phenomenons and are used in many applications, such as, meteorological readings, price stock readings, ECG signals, and any data collected by a sensor over time.

In anomaly detection scenarios, a crucial characteristic for the analysis of time series is the periodicity, which is the characteristic of a segment of a time series to repeat itself over time. Mathematically it is represented as $X(i) = X(i + T)$ in which T is the period of the time series. Hereafter, for simplicity reasons, quasi-periodic time series, which are time series that are repetitive in essence but may have morphology differences in different "cycles", will be referred to as periodic or cyclic.

2.2 Anomaly on Time Series

Anomalies are usually defined as individual points or groups of points that do not conform with the expected behaviour of the whole data set [4]. Given the critical importance of finding anomalies described earlier, it is extremely important to categorise the

existing anomaly types:

Point Anomaly - An individual point that does not conform well to the whole data set. It is presented an example in Figure 2.1a.

Sequence Anomaly - A sequence of points that does not fit well with the rest of the data set. Example in Figure 2.1b.

Pattern Anomaly - A short segment of data that forms a pattern, but does not conform well to the pattern formed by the rest of the data set. Figure 2.1c shows a simulated pattern anomaly.

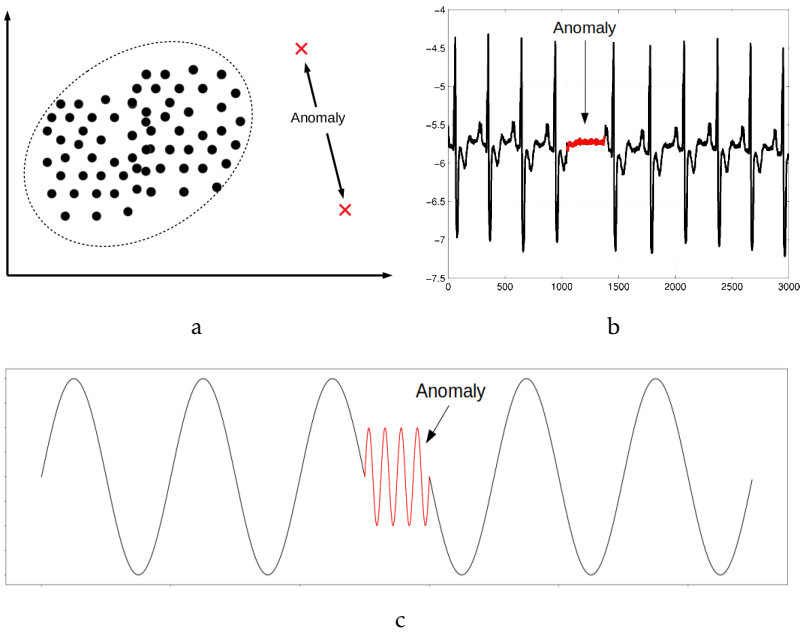

Figure 2.1: Representation of each anomaly type: a) point anomalies; b) sequence anomaly. Adapted from [1]; c) pattern anomaly.

The main goal of this work is to detect anomalies in periodic time series and, therefore, the principal focus will be finding sequence anomalies and anomalous patterns.

There are various difficulties in the process of anomaly detection and several characteristics that have to be taken into account in order to perform accurate anomaly detection. The first problem that must be taken into account, which is present in most data mining settings, is noise. Noise can mask not only the normal behaviour of a time series, but also the anomalous. For example, in Figure 2.2, noise is so relevant that it is almost impossible to perceive the behaviour of the original signal.

The second characteristic to consider is the context. Once anomalies are considered highly context dependent, signals acquired in different domains may have different concepts of anomalies, such as different concepts of normality. Thus, there are points in a signal that are not anomalous if considered without context, but if we add it, their presence indicates an anomaly. For example, if we consider temperature readings over a year

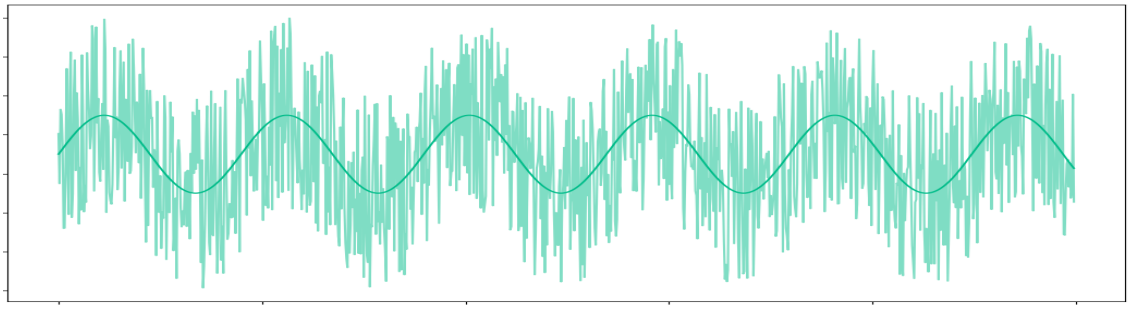

Figure 2.2: Noise influence on a time series. The darker signal can be masked by the existence of noise, represented by the lighter part, making it difficult to find anomalies or learn the normal behaviour of a time series.

in Portugal, it is normal to find a temperature of 10ºC. However, if this reading is made in the summer, it may be considered anomalous, because it is not usual to have these temperatures by that time of the year. An illustrative example is shown in Figure 2.3.

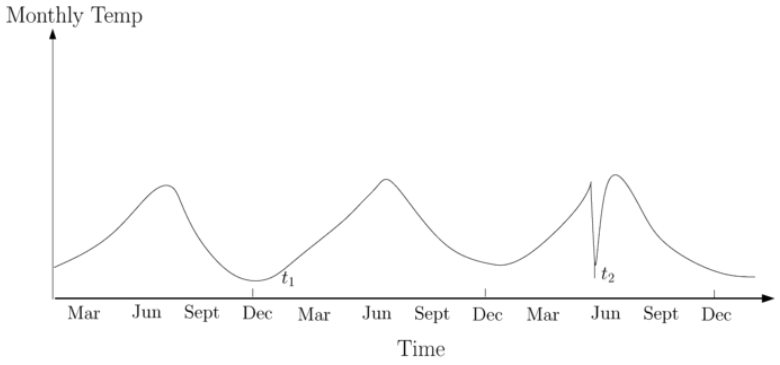

Figure 2.3: Representation of a contextual anomaly. The temperature is an example of how the context may influence the task of finding anomalies. Here we see that there are other points, such as t_1 with the same value as t_2, yet only t_2 constitute an anomaly. Image source: [1]

The last characteristic to consider is the frequency of anomalies. There are algorithms that make the assumption that the anomalous sequences or patterns appear less often than the normal data points, however, it might not always be true. For instance, the presence of a worm in computer networks is reflected by a higher amount of anomalous than normal traffic [1].

Moreover, there are various challenges in anomaly detection on time series [15]. The first, is the fact that the portion of the time series that is anomalous is unknown and can be a single point (point anomaly) or a sequence that may correspond to a cycle in a cyclic time series, it may occupy a portion of a cycle or even various consecutive cycles, ultimately reaching the whole time series. Besides, once the morphology of anomalous segments is unknown, it is difficult to select an appropriate distance measure or function to test different signals of different domains. Furthermore, in multivariate time series, it might happen that anomalies exist in a few variates, instead of in all, which complicates

the task of anomaly detection in such cases.

Despite the characteristics and challenges that have to be taken into account for anomaly detection, it is possible to formalise the general problem of anomaly detection:

Given a time series $X = \{x_1, x_2, ..., x_N\}$, we can segment the time series in M subsequences as

$$X = \{S_1, S_2, ..., S_M\} \tag{2.2}$$

where each $S_i, i \in \{1, 2, ..., M\}$ is a subsequence of X composed of a defined number of data points, that may vary from segment to segment. So the time series can be represented as

$$X = \{S_1, S_2, ..., S_M\} = \{\{x_1, ..., x_{k_1}\}, \{x_{k_1+1}, ..., x_{k_2}\}, ..., \{x_{k_{M-1}+1}, ..., x_{k_M}\}\} \tag{2.3}$$

where $k_i, i \in \{1, ..., M\}$ are the lengths of the corresponding segments, S_i, of X.

The analysis of each subsequence is usually accomplished using a cost function that may, for instance, indicate distance or density. Thus, a subsequence S_i is anomalous if

$$func(S_i, S_j) > \delta \quad \forall j \in [1 : M] \tag{2.4}$$

where S_j may correspond to all subsequences except S_i, a model of a normal pattern, or a set of rules that S_i must obey to be considered a normal segment. The result of $func(S_i, S_j)$ is the anomaly score, which can be considered the *anomaly degree* of S_i. The definition of the threshold, δ, usually controls the sensitivity of the algorithm.

There are numerous methods used to detect anomalies. In the next sections there will be described the most utilised techniques followed by a brief summary and discussion about them.

2.3 Statistical Methods

This class of techniques is based on one principle: the process that controls the underlying behaviour of a data set is a stochastic event, thus, it follows a statistical distribution. Thereafter, an anomaly is originated by a different process that does not follow the same statistical distribution [1]. The key assumption is that normal instances populate high probability regions, while anomalous data is located in low probability regions of the assumed stochastic model. These techniques are divided in parametric and non-parametric.

Parametric techniques assume that the normal behaviour of a data set is generated by a parametric statistical distribution. Hence, given an observation x, there is a set of parameters θ that parametrise the probability density function $f(x, \theta)$. The parameters are estimated from the observed data and the anomaly score is the inverse of the probability density function $f(x, \theta)$. Therefore, an observation with higher probability of appearance has an inferior anomaly score than an observation with a lower probability value.

One of the most used techniques among parametric techniques is the Gaussian Model-Based, which assumes that the data is generated accordingly to a Gaussian distribution.

The parameters are estimated using *Maximum Likelihood Estimates* (MLE) and the anomaly score is often calculated as the difference between the test instance and the parameters, namely the mean and standard deviation, of the assumed probability density function. There are more relevant statistical tests which can be used in anomaly detection techniques and the assumed probability density function can also be changed depending on the application.

Another widely used technique is the Regression Model-Based, in which a regression model is firstly fitted to normal data. Then, for test instances, the residual is used to calculate the anomaly score.

Contrary to parametric techniques, non-parametric techniques do not assume data distribution and use non-parametric statistical models, hence, the model structure is determined by the analysed data. These techniques make fewer assumptions than parametric techniques. The most basic technique is the Histogram Based, in which the profile of the instances are used to compare them.

The greatest advantage of these techniques is that if the assumed distribution is true, it is possible to justify the classification of each instance and it is possible to make a statistical study of each result, resulting in a robust classification. Otherwise, the greatest disadvantage is that if the assumed distribution does not hold true, the statistical tests are not valid. Furthermore, there are various distributions that may be applied to the same scenario and all of them may be true in some point, which makes it difficult to choose the appropriate distribution.

2.4 Forecasting Methods

Forecasting methods rely on the apprehension of the normal behaviour of a time series based on its past in order to be able to accurately detect future anomalies. Hence, the first step is to learn the normal behaviour based on the history of the time series and it is critical to choose the correct time-interval to be considered history. This is because, in cyclic events, if the length of the signal that is being learned is inferior to the time of a repetitive unit, the algorithm will not be able to accurately learn the shape of a normal cycle and, thus, it may have a high false hit rate. If the considered history is too long, then the computational complexity can play a role in a way that it might be impossible to analyse the whole data set [15]. In Figure 2.4 there is an example of time series forecasting, in which the blue part is the considered history in order to make the predictions presented in green. The anomaly detection task, which is the second step, consists on the comparison of the predicted model and the true observations, being the anomaly score given by the difference between those. In Figure 2.4, the true observations correspond to the orange part, that would be compared to the green part if the task was to detect anomalies.

There are forecasting methods that are *time series based* and methods that are *non-time series based* [15]. Time series based techniques use the whole time series data as well as

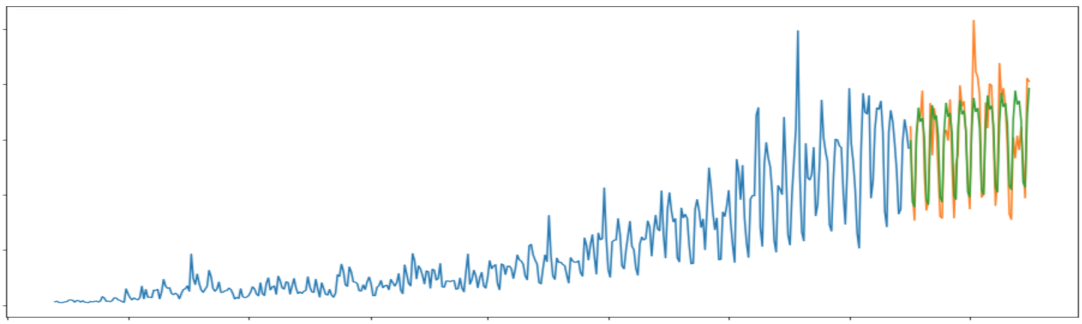

Figure 2.4: Time series forecasting example. The blue part is the history considered to make the predictions, the green part is the achieve prediction and the orange part is the real signal at that time interval for comparison. Image source: [33]

the history parameter m, which is the number of points of the time series to be considered as history and is considered the order of the model, in order to make predictions. There are various models that may be used:

- **Moving Average (MA)** - MA models represent time series by the application of a linear filter, in which each point is represented by the mean value of the m points surrounding it. This means that the time series appears to be smoother. The forecast consists on taking the last m points of the time series and assuming that the next point will have an identical value (it may be considered the addition of noise):

$$y_{MA}(t+1) = \mu + \epsilon_{t+1} + \sum_{i=0}^{t-m} \theta_{t-i} \cdot \epsilon_{t-i} \tag{2.5}$$

  where θ_{t-i} are the model's coefficients, ϵ_{t-i} are white noise terms of past observations, ϵ_{t+1} is the white noise term of the future observation and μ is the mean of the time series.

- **Autoregressive (AR)** - AR models represent each series data points as regressions of previous data. The predicted data is the result of the regression of m previous observations and is calculated according to Equation 2.6:

$$y_{AR}(t+1) = \sum_{i=0}^{t-m} \beta_i \cdot y(t-i) + \epsilon_{t+1} \tag{2.6}$$

  where β_i is the i^{th} autoregressive coefficient and ϵ_{t+1} is noise at time $t+1$. It is similar to linear regression, but instead of being only dependent on the current observation, the result is also dependent on previous results.

Autoregressive Moving Average (ARMA) models combine both AR and MA to produce the prediction. It is modelled by Equation 2.7 which is the sum of equations 2.5 and 2.6

$$y_{ARMA}(t+1) = \mu + \epsilon_{t+1} + \sum_{i=0}^{t-m} \theta_{t-i} \cdot \epsilon_{t-i} + \sum_{i=0}^{t-m} \beta_i \cdot y(t-i) \tag{2.7}$$

Another widely known method is the Autoregressive Integrated Moving Average (ARIMA) which works based on the same principle as ARMA, but in the differentiated signal, in which each data point corresponds to the difference between the actual point and the previous one.

Non time series based techniques take only into account the information given by the history of the m data points from the recent history of the time series. These techniques, such as Linear Regression and Gaussian Processes would have the form of

$$y = W \cdot \Phi(\{x_{t-m}, ..., x_t\}) + b \tag{2.8}$$

where $\{x_{t-m}, ..., x_t\}$ are the previous m observations, W is a weight vector and Φ is a mapping function (in linear regression, $\Phi(\{x_{t-m}, ..., x_t\}) = \{x_{t-m}, ..., x_t\}$).

The greatest advantage of these techniques is that if the underlying process is continuous and dependent on past states, the prediction may be accurate, and so the anomaly detection step is also correct. But if the process is not recursive and the next value does not depend on past values, then the predictions made are inaccurate and anomaly detection becomes impossible.

2.5 Machine Learning

Machine Learning consists of a set of methods capable of automatically detect patterns in data and then use the discovered patterns in decision making scenarios and to predict future data [34]. Ultimately, the goal is always to generalise the learned pattern for any input data, and to do so, there is one assumption that is made: If two points x_1 and x_2 are similar, then so should be the corresponding outputs y_1 and y_2.

In Figure 2.5 there is a typical pipeline of application of machine learning techniques, in which the top part corresponds to the supervised learning workflow and the bottom part corresponds to the unsupervised learning methodology.

Those workflows are made accordingly to the supervision degree required to learn the referred patterns, that are:

Supervised learning - In this type of learning the goal is to apprehend the behaviour of a dataset given a train set where each data point is labelled and categorised. A set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where x_i is a data point, y_i is its label, $\mathcal{D}$ is called training set and N is the number of training examples, represents a possible representation for a data set used in supervised learning. Likewise, there is a test set which has unlabelled data that can be different from the training set. Given the training set, if a point from the testing set is given as an input, the model will be able to label it with one of the labels $y_i, i \in \{1, ..., N\}$ present in the training set. Until now, for simplification purposes, we have been considering x_i as being a data point, but it may be a set of points such as a vector or matrix. This is extremely useful when dealing with multivariate data sets, for instance the height and weight of a group

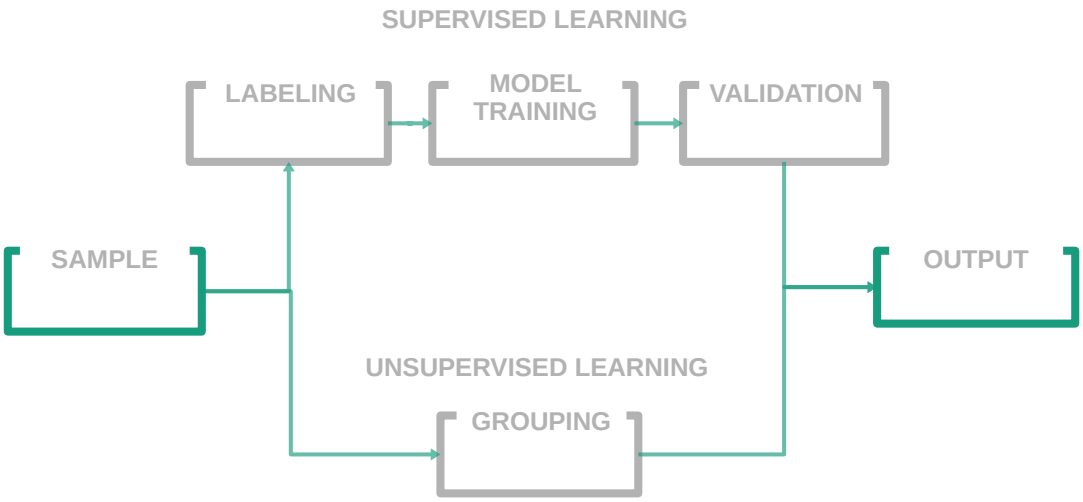

Figure 2.5: Typical pipeline for machine learning techniques. On top there are the procedures taken by supervised learning techniques and in the bottom is the procedure for unsupervised learning techniques.

of people, images, time series data or sentences. The label or output may also have different designations. It can be considered categorical, if it has a finite number of possible values $y_i \in \{1,...,C\}$, in which case, the task is called classification, or it can be a real-valued scalar $y_i \in \mathbb{R}$ being then called regression. One variant of this is ordinal regression, consisting of a categorical output that has a natural order, such as any grade system [34]. The most known example of supervised learning classification algorithm is the k-nearest neighbour. Given a labelled dataset and a testing data point, the label attributed to the test point is the most common label among its k nearest neighbours, which are the k closest points to the test point in the training set.

Semi-supervised learning - In this type of learning the goal is to apprehend the behaviour of a data set given a train set in which there are labelled and categorised data points and data points that are not labelled [35]. Formally, X is the training set containing N inputs, that contains $X_s = \{(x_i, y_i)\}_{i=1}^{l}$ and $X_u = \{x_i\}_{i=l+1}^{N}$ being X_s the set of labelled inputs and X_u the set of unlabelled inputs. There is a variation of the classic semi-supervised learning, which is unsupervised learning guided by constraints, where there are no labels, but the system receives information to cluster certain data points in the same category. For semi-supervised learning to work, it is necessary that the training set fulfils a critical prerequisite, which is that the distribution of labelled and unlabelled data has to be relevant. In anomaly detection, the distribution has to be: either the normal data is labelled *or* the anomalous

data is labelled, instead of mixing the nature of the labelled data. This means that the (un)labelled data carries information to answer the question $p(y|x)$ (what is the probability of the output to be y given that the input is x). If we consider boolean data sets, the process is simple because we could label every data from a class, thus, every sample that does not belong to that class, belongs to the other. In anomaly detection, for example, it is common to have a training data set with the normal instances labelled. This is helpful in situations where there are few or none anomalies in the training set. For example, in spacecraft fault detection, an anomaly scenario is not easy to model because it would implicate an accident [35]. The typical approach used in such techniques is to build a model for the class corresponding to normal behaviour, and use the model to identify anomalies in the test data [1]. The opposite case, the training set containing only anomalies is difficult to work with because anomalous instances may have significantly different morphologies and they may be unknown for most contexts.

Unsupervised learning - In unsupervised learning the goal is to apprehend the behaviour of a dataset without prior knowledge about it [34]. Thus, all data is unlabelled at the beginning and $X = \{x_i\}_{i=1}^{N}$ is a possible representation of the inputs to unsupervised learning techniques. The goal is to discover interesting structures that can divide the data instances into clusters of related data, even if that relation is unknown, such that inner cluster variance is minimum and inter cluster variance is maximum. Clustering algorithms may be divided in exclusive or non-exclusive algorithms. In exclusive algorithms each sample is assigned to a specific cluster, while in non-exclusive algorithms, each sample may belong to more than one cluster. Exclusive algorithms include partitioning, hierarchical and density-based algorithms. Partitioning, in which it is included, for example, k-means and k-medoids, clustering is performed based on the distance to a specific point, corresponding to the centroid of the cluster. In k-means, that point is the gravity-center of each cluster, while in k-medoids it is the data point closest to the center. Each data point is thus assigned to the cluster with closest centroid or medoid. Hierarchical methods construct a dendogram based on the similarity between samples in order to form clusters. The difference between methods is the way in which the dendogram is constructed (top-down or bottom-up) or the similarity measure. Density-based algorithms partition data points based on the density of each data point. Points belonging to dense spaces are considered clusters and points sparser in space are considered outliers. Non-exclusive algorithms, such as C-means or Gaussian Mixture Models use the concept of probability and fuzzy analysis in order to group data points. In C-means it is used a fuzzy approach in order to assign a probability of belonging to a given cluster to each data point, which means that it may belong to various clusters with different probabilities. Gaussian Mixture Models assume that each cluster has the form of a Gaussian Function and each data point is assigned to a probability of

belonging to each cluster based on the spatial location relative to the center of each probability-density function. In Figure 2.6, there is an example of various clustering algorithms applied in various datasets in order to comprehend the difference of their application.

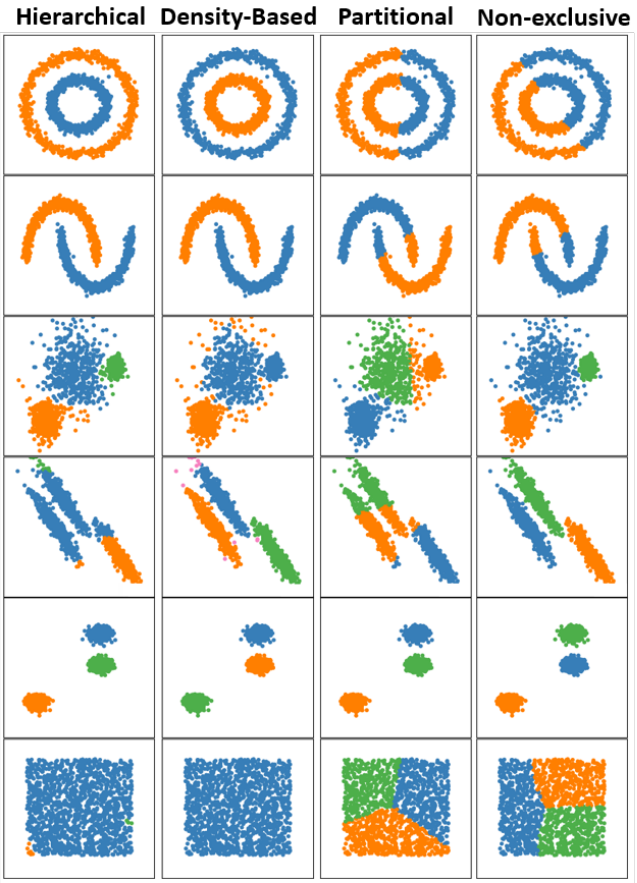

Figure 2.6: Clustering algorithms exemplified. Adapted from: [36]

Unsupervised learning is more used than supervised, because it does not require manual labour to label any data set, which is a process that might be slow, may be expensive and may hide important information that a human might discard without realising. For example, retailers who track customers purchases use this information to detect groups of customers with similar interests or similar buying patterns and can then use this for marketing purposes, among other applications [37].

Machine Learning has been used in several applications such as email spam filtering, image classification, prediction of future's stock market price given current market conditions and signal processing [34].

The main advantage of machine learning techniques is that, if correctly trained, the algorithms may be applied for previously unseen data, making accurate predictions or classification due to the generalisation that they are able to demonstrate. The main disadvantage is that the training process may be time consuming, data labelling may be

tedious, require an expert to do it, and it may be necessary a great amount of labelled data in order to train supervised algorithms. Furthermore, even though unsupervised algorithms, such as, clustering algorithms, do not require labelled data, it is necessary to understand the outcome of those algorithms, which might not be simple.

2.5.1 Time Series Metrics - Features

In machine learning it is usual to represent each sample by specific characteristics, designated as features. This process is usually designated as feature extraction and the task is to extract the most relevant features to characterise the analysed samples. Relevant features are defined as features that allow the distinction of samples in different classes, namely, the classes that we are trying to assign them to, but are not *redundant* or *irrelevant*. Redundant features are features with high values of correlation, which means that they carry similar information, while irrelevant features are those which are not able to distinguish samples of different classes, usually because the range of values that they take is narrow. Feature selection is the step in which both irrelevant and redundant features are eliminated and allows the simplification of models, shorter training times, reduced overfitting (thus increased generalisation) and to avoid the curse of dimensionality. The curse of dimensionality arises from increased number of variables, because with that augmentation the number of configurations of the data rises exponentially [38]. In Figure 2.7 it is clear that an increase of dimensionality (left to right) leads to more complex solutions, the number of regions of interest increases and the task of generalisation becomes harder because of the specificity of those regions. Thus it is important to select appropriate features depending on the domain of application.

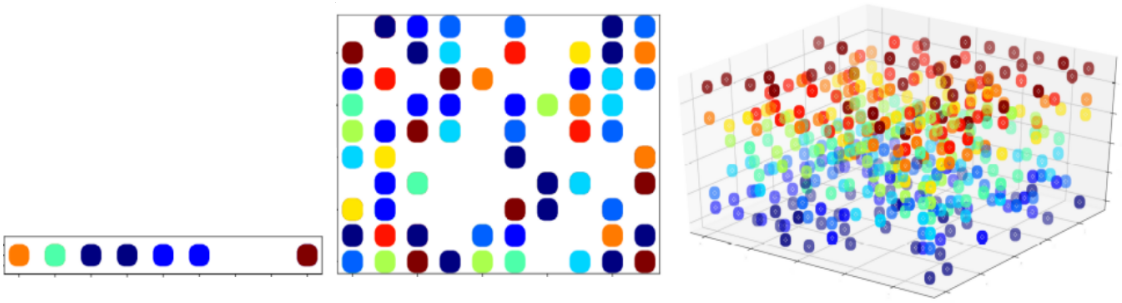

Figure 2.7: Curse of Dimensionality illustrated. In the left image it is easy to distinguish between different areas of interest, in which each area is represented by a cell. As the number of dimensions is increased (left to right), the number of regions of interest increases and the generalisation process becomes harder, because each region becomes more specific.

Furthermore, it may also be crucial the weighting associated to each feature. If in a specific application a feature is in the order of the thousands and other feature in the order of units, then the first feature will dominate the result because of its weight. Thus, it is

important to scale the features correctly, but that task is not trivial. An extensive research of methods that help to address that problem is given in [39]. In cluster analysis there are three widely known scaling procedures: Z-score standardisation, range standardisation and variance-to-range standardisation.

- **Z-score standardisation** - each variable is multiplied by the reciprocal of the standard deviation. Hence, each feature is left with unit variance, which means that features with higher variance are more penalised than features with low variance. The formula used for this scaling is presented in Equation 2.9

$$z_{ij}^{(1)} = \frac{x_{ij} - \bar{x}_j}{\sigma_j} \tag{2.9}$$

  where $\bar{x}_j$ is the mean value of the set and σ_j is its standard deviation.

- **Range standardisation** - each variable is multiplied by the reciprocal of the range of the feature. The result is that every feature is left with unit range and if it is centred around zero (zero mean), the resulting range of values is from $-0,5$ to $0,5$. Thus, every feature has equal contribution to the result. The transformation is made by 2.10.

$$z_{ij}^{(2)} = \frac{x_{ij}}{max(x_j) - min(x_j)} \tag{2.10}$$

  where $max(x_j)$ and $min(x_j)$ are the maximum and minimum value of each feature set, respectively. In this case the value of variance can be different for each feature.

- **Variance-to-range ratio weighting** - this scaling technique takes into account both variance and range information. This type of scaling is appropriate for clustering techniques because it gives a general idea about the *clusterability* of each feature. According to [39], there are three steps to calculate the appropriate value, starting by the calculus of M_j by Equation 2.11.

$$M_j = \frac{12 \cdot Var(x_j)}{max(x_j) - min(x_j)} \tag{2.11}$$

  where M_j is an indicator of the clusterability of x_j, increasing with that characteristic. Relative clusterability is given by Equation 2.12.

$$RC_j = \frac{M_j}{min(M_j)} \tag{2.12}$$

  A set of features with no clusterability will have RC=1 and the increase of clusterability leads to an increase of RC. The transformation is made using Equation 2.13.

$$z_j^{(3)} = z_j^{(1)} \cdot \sqrt{\frac{RC_j \cdot [max(z_{min}^{(1)}) - min(z_{min}^{(1)})]^2}{[max(z_j^{(1)}) - min(z_j^{(1)})]^2}} \tag{2.13}$$

where $z_j^{(1)}$ is the matrix of the transformation of the x_j variable and $z_{min}^{(1)}$ is the transformation corresponding to the value with minimum M value calculated by equation 2.11.

Features scaling can be made prior or after feature selection, and the used method to achieve it is usually chosen empirically by experimentation and analysis of results. The sensitivity towards features extraction and selection represents another disadvantage of machine learning methods, because their selection may have a high influence on the achieved outcome.

2.5.2 Validation

Validation consists on the evaluation of the results achieved in the classification or clustering phase. For that, it is required the definition of various concepts that characterise the classification of each sample. First, the expected outcome corresponds to the true class of the classified sample, whereas the predicted class corresponds to the classification given by the model which may be supervised, semi-supervised or unsupervised. In the case of anomaly detection there are two classes: normal and anomalous. Based on these concepts, it is possible to define four quantities to assess the classification:

True Positives (TP) - number of samples classified to the same class as the expected class. In anomaly detection it corresponds to assigning the label of anomalous to an anomalous sample.

False Positives (FP) - number of samples classified to a different class in relation to the expected class. In anomaly detection it corresponds to label normal samples as anomalous.

True Negatives (TN) - number of samples that do not correspond to the class that is under consideration classified as different from that class. In anomaly detection scenarios, it corresponds to label normal sample as normal.

False Negative (FN) - number of samples that correspond to the class that is under analysis classified as different from that class. For example, labelling an anomalous sample as normal.

This quantities are usually expressed in the form of a confusion matrix as shown in Table 2.1.

Table 2.1: Confusion matrix representation.

	Expected Label	
Predicted Label	Negative	Positive
Negative	TN	FN
Positive	FP	TP

Moreover, it is possible to calculate diverse metrics in order to clearly compare the results achieved by different processes. Those metrics are expressed in Table 2.2.

Table 2.2: Validation metrics.

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Sensitivity	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F1 score	$\frac{2 \cdot TP}{2 \cdot TP+FP+FN}$
False Negative Rate (FNR)	$\frac{FN}{TP+FN}$
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
False Discovery Rate (FDR)	$\frac{FP}{TP+FP}$
Negative Predictive Value (NPV)	$\frac{TN}{TN+FN}$

Accuracy is the percentage of correctly classified samples, specificity is the percentage of correctly classified negative instances (percentage of normal instances classified as normal in the case of anomaly detection), sensitivity is the percentage of positive samples classified as positive (percentage of anomalous instances classified as anomalous in the case of anomaly detection) and precision is the percentage of positives that are actually positive (percentage of anomalies that are in fact anomalies). F1 score is mostly used for unbalanced data sets, in which the number of instances per class is very dissimilar, such as the case of anomaly detection, where the number of anomalies is usually inferior to the number of normal samples, and is the harmonic mean of sensitivity and precision. NPV is the corresponding metric to negative instances as precision is to positive samples. Furthermore, FNR measures the amount of misclassified positive instances (percentage of anomalies classified as normal instances), FPR is the corresponding of FNR for negative instances (percentage of normal samples classified as anomalous), FDR measures the amount of incorrectly classified positive instances (percentage of incorrectly classified normal instances considered anomalous among all classified as anomalous). NPV corresponds to the percentage of correctly classified negative instances among all classified as negative.

2.6 Summary

In this chapter there were introduced the concepts needed to understand the anomaly detection task, as well as generic approaches that have been used for that task. It was given a definition for time series and for anomaly on time series, which was important for the work developed during the course of this dissertation.

It was shown that statistical methods for anomaly detection have the advantage of being able to justify the results, but have a great disadvantage, which is that if the assumed probability distribution is not true, then the statistical analysis does not have significance and might be inaccurate. Furthermore, forecasting methods may be relevant for anomaly detection but are highly dependent on the considered history. Machine learning methods may apprehend the normal behaviour of a given signal, but are greatly influenced by feature extraction, selection and normalisation, which might represent a problem, but may also find hidden structures that are not easily distinguishable for the human eye.

Chapter 3

A framework for anomaly detection on generic repetitive Time Series

Given the formulations required for a comprehensive view of the problem of anomaly detection, it is now possible to describe the developed framework designed to tackle this problem.

3.1 Requirements and Overview

It is proposed, with this framework, the detection of anomalies on generic repetitive time series, independent of the domain or context of the signal. In order to aspire for a generic application, it is necessary to establish assumptions regarding the type of anomalies that will be detected.

There are three assumptions that are made by the proposed anomaly detection framework:

- **Repetitiveness** - The signal is repetitive or it exhibits near repetitive behaviour.
- **Type of anomaly** - The signal may have all types of anomalies defined in Section 2.2, but the framework only detects sequence or pattern anomalies.
- **Frequency of anomalies** - There are always less anomalous instances than normal instances.

The proposed approach is illustrated in Figure 3.1 and starts with data acquisition followed by an unsupervised segmentation algorithm used to extract each cycle, which corresponds to the repetitive unit of quasi-repetitive signals, that may be normal or anomalous. Then, there is a step of feature extraction from each extracted cycle. Those features will be the input for a clustering algorithm that will perform unsupervised anomaly detection.

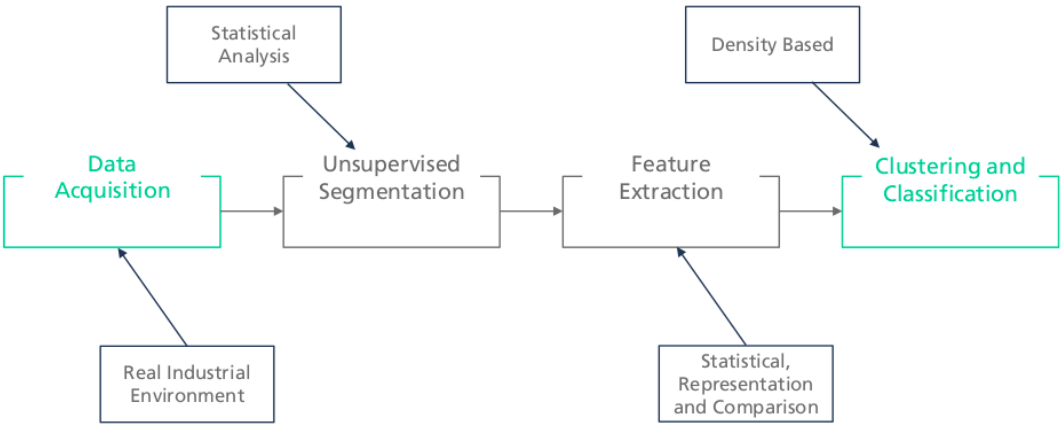

Figure 3.1: Proposed framework for anomaly detection on time series.

3.2 Unsupervised Segmentation

In order to segment generic repetitive time series with no *a priori* knowledge, it is necessary to use an unsupervised segmentation algorithm. The algorithm must be able to extract every cycle from different domains due to the fact that, in manufacturing environments, different workstations have different concepts of normality, and so, generate different cyclic signals. Those differences manifest through the morphology and period of the signals. Thus, the algorithm must be dictionary-free (see section 1.3.1), because it is impracticable to have models to each workstation. This requirement also allows to design a solution that might be applied in different domains, as long as the signal is quasi-periodic.

This challenge was overcome by the development of a new unsupervised segmentation algorithm, which consists of two parts based on a statistical analysis. For the first part, the base assumption is that in a perfect cyclic signal, all cycles are equal, thus, the mean value of every cycle is also identical. First, we represent a cyclic signal by the set of means of the cycles, according to Equation 3.1, where x_{ij} is the i^{th} value of the segment j and k_j is the number of data points of that segment.

$$\bar{S}_j = \frac{\sum_{i=0}^{k_j} x_{ij}}{k_j} \tag{3.1}$$

Then, it is possible to calculate the standard deviation of those means, using Equation 3.2, where M is the number of cycles, $\bar{S}_a$ is the mean value of the set of means of all segments and a is the iteration.

$$\sigma_a = \sqrt{\frac{\sum_{j=1}^{M} (\bar{S}_j - \bar{S}_a)^2}{M-1}} \tag{3.2}$$

These metrics are trivial, but it is important to have in consideration the indexes of every variable in order to have a clear perception of every step. Thus, once the mean of each

cycle is identical, each value in the set of means is equal to the average value of the set of means, leading to a standard deviation of 0 ($\bar{S}_1 = \bar{S}_2 = ... = \bar{S}_M = \bar{S}_a \implies \sigma_a = 0$). Figure 3.2 shows the described approach.

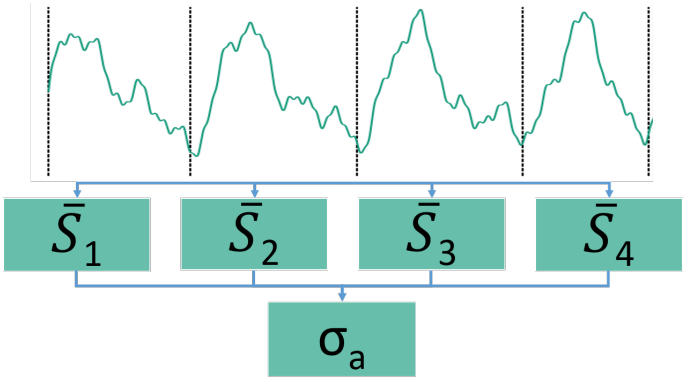

Figure 3.2: Representation of each segment in terms of its mean value followed by the representation of the iteration by the standard deviation of the set of means.

The proposed algorithm starts by dividing the time series in five parts. The parts may have different lengths because their limits are forced to be local minimums in a user-specified range. This segmentation is performed iteratively increasing the number of segments in each iteration, as shown in the left pane of Figure 3.3. Then, in each iteration, it is calculated the average value of each segment and each iteration is represented by the standard deviation value of the set of means. This forms a function of iteration *vs.* standard deviation, which allows the visualisation of a plot as the one in the right pane of Figure 3.3.

The second part starts by the selection of the iterations corresponding to local minima (positive second derivative) in the function of iteration *vs.* standard deviation. These iterations correspond to segmentation results in which the similarity between segments increases. Then, for each of those iterations, it is computed the *Pearson's correlation coefficient* (PCC) between each segment S_j and all the others by Equation 3.3, where $j, m \in \{0, 1, 2, ..., M\}$, $j \neq m$, $\sigma_{S_{l,l\in\{j,m\}}}$ is the standard deviation value of each segment and where the covariance is calculated by Equation 3.4, in which $E[*]$ represents the mean of its input.

$$\rho_{S_j,S_m;a} = \frac{cov(S_j, S_m)_a}{\sigma_{S_j}\sigma_{S_m}} \tag{3.3}$$

$$cov(S_j, S_m)_a = E[(S_{ij} - \bar{S}_j)(S_{im} - \bar{S}_m)] \tag{3.4}$$

Thus, each segment is now represented by the mean of the PCCs, $\bar{\rho}_{S_j;a}$, to all other segments. This allows us to infer about the similarity of each segment in relation to the rest. PCC is restricted between -1 and 1, where 1 corresponds to the value of two directly correlated vectors, 0 to two non-correlated vectors and -1 to two inversely correlated

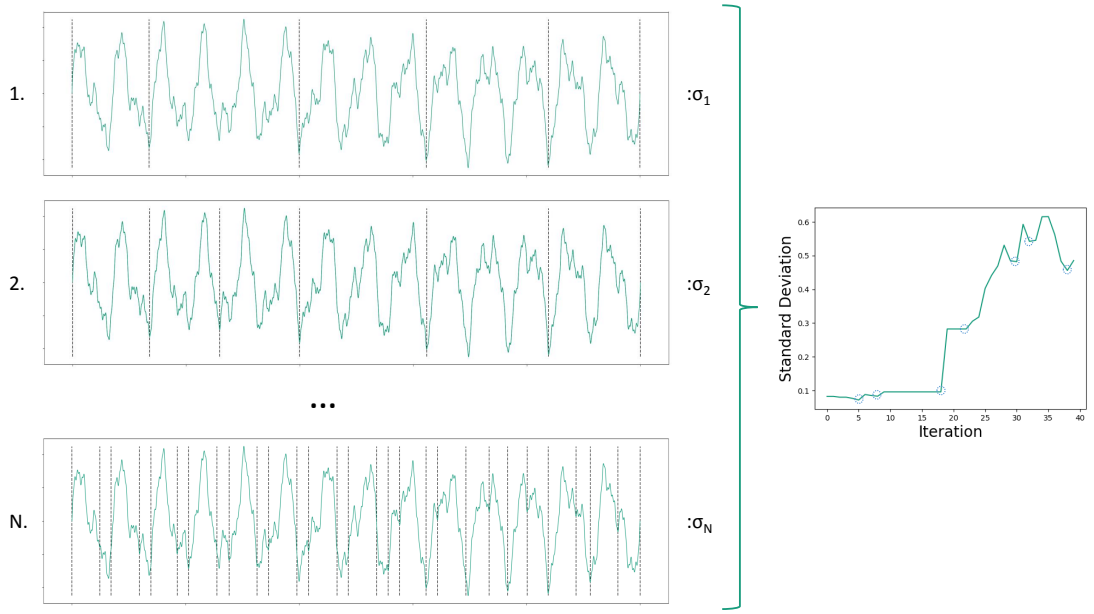

Figure 3.3: Top-down process of segmentation. Firstly, the time series is segmented in 5 parts. Then, with each iteration, the number of segments increases and in each iteration the mean of each segment is computed. Each iteration is represented by the standard deviation of the set of means of its segments forming a curve such as in the image in right. The negative inflexion points are chosen for the rest of the process.

vectors. Thus, the closer to 1 is the value of PCC, more correlated are the two segments, and the more similar are the compared vectors. The segmentation is then represented by the mean value of the mean correlation value of every segment to all others (see Figure 3.4):

$$\bar{\rho}_a = \frac{\sum_{j=0}^{M_a} \bar{\rho}_{S_j}}{M_a} \tag{3.5}$$

where M_a is the number of segments of iteration a. Hence, the value that represents each segmentation iteration is restricted between -1 and 1. The selection of the correct segmentation is based on this value and corresponds to the iteration with the highest value, meaning that all segments have high correlation values to all others.

Despite all this, it is needed to notice that the covariance matrix is a square matrix, which means that the signals used to construct it have to be of the same dimension. In the developed framework this is made by extrapolating all segments to the dimension of the largest one, prior to all computation. Furthermore, while the selection of the segmentation to maintain is restricted to the second part of the algorithm, it is necessary to point that the first part of the algorithm is essential to reduce the time required for the segmentation, because the calculation of all PCCs is (much) more time consuming than calculating the set of means followed by its standard deviation. Therefore, while being dispensable, the first part of the algorithm is critical to increase the algorithm's performance.

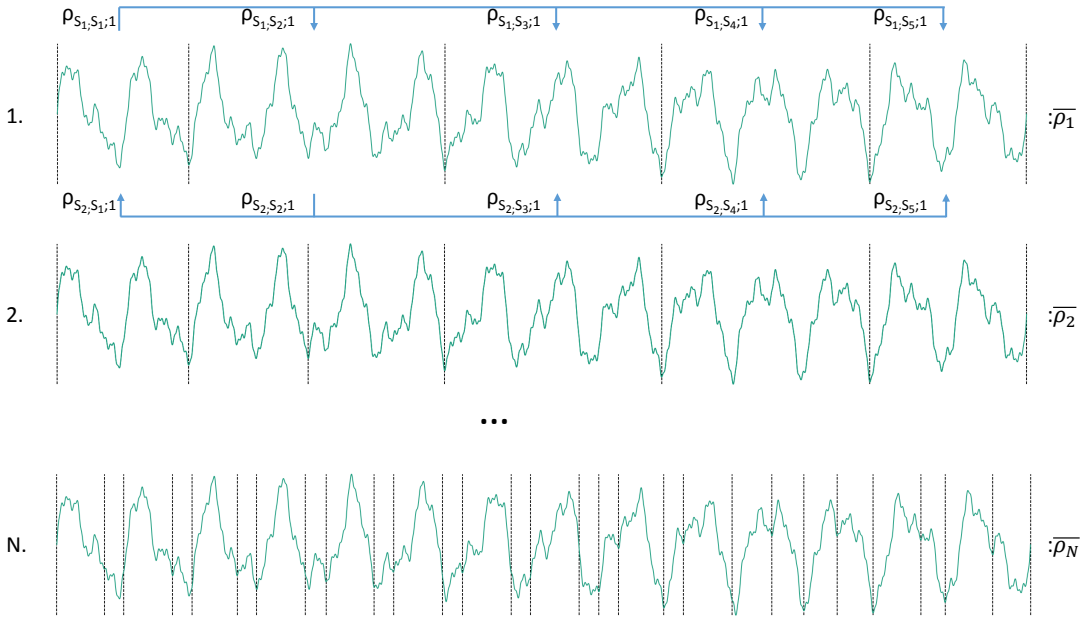

Figure 3.4: Second part of the unsupervised segmentation algorithm. For every iteration selected from the first part of the algorithm, it is computed the mean value of the correlation value for each segment to all others. Then, it is calculated the mean value of those means for each iteration, which is then represented by that value. The iteration that has the highest value of its representation is then selected as the correct segmentation.

For multivariate time series, such as signals acquired with resource to inertial sensors, where multiple sensors acquire data from the same process in a three dimensional way, the segmentation is performed for each direction of each signal and the integration is made by comparing the values of correlation between every signal, again being chosen the signal with highest $\bar{\rho}_a$. The indexes of the segmentation of that signal are, then, propagated for the rest of the signals and the segmentation is performed by those indexes.

3.3 Feature Extraction

After segmenting the cyclic signal, it is necessary to represent each segment/cycle in a concise way that is able to distinguish between classes, which in this case are normal and anomalous. In order to achieve this, relevant features have to be extracted from each cycle. These features are characteristics that can capture the individual behaviour of each extracted cycle that are considered as samples, as described in Section 2.5.1.

With this purpose a set of features was chosen and their selection is one of the most important parts of the process because while there are features that can characterise the normal cycles and distinguish them from the anomalous ones, there are other features that are unable to differentiate the classes. To achieve the intended separation, several features were tested and applied in this framework, and are enumerated in Table 3.1.

Table 3.1: Utilised features for representing each segment of time series data.

Statistical Features	Representation Transforms	Comparison Metrics
• Mean Value • Standard Deviation • Minimum Value • Maximum Value • Inter-Quartile Range • Number of Peaks • Median • Kurtosis • Skewness • Duration • Linear Regression • Zero Crossing Rate • Polarity • Cumulative Summation • Histogram	• Fourier Transform • Wavelet Transform • Principal Component Analysis Transform • Independent Component Analysis Transform • PAA in the Amplitude Domain (AD-PAA) • PAPR • Subsegment Analysis	• Euclidean Distance • Dynamic Time Warping Distance • Time Alignment Measurement • Pearson's Correlation Coefficient • Cosine Similarity

There were used *statistical features*, that attempt to capture differences of the underlying statistical model which origins the cycles, assuming that the normal cycles are originated by a given statistical model, while anomalous segments are assumed to be generated by a different statistical model; *representation transforms*, which attempt to capture characteristics that may not be simple to find in the raw signal; *distance metrics*, that rely on the comparison between segments or to a given template.

3.3.1 Statistical Features

Statistical features attempt to capture the behaviour of the probability density function of the analysed instances. It is considered that normal instances are generated from a similar probabilistic distribution characterised with similar parameters, while anomalous instances are generated from different distributions or different parameters.

The most known features are the mean, standard deviation, minimum, maximum and median value, which characterise the probabilistic density function that underlies the process that generated the instances.

Inter-quartile range (IQR) is the value corresponding to the difference between the third and first quartiles of a given statistical distribution. Quartiles are the values that separate a given Gaussian distribution in four equally populated parts. Thus, there are three quartiles Q_1, Q_2 and Q_3. Q_1 is the value bellow which exist 25% of data points, Q_2 is the value bellow which there are 50% of data points and above Q_3 there are 25% data points. Thus, between the first and third quartiles there are 50% of data points. The calculus of these can be made by the integration of a given probabilistic density function from $-\infty$ to $0,25$ for Q_1 and in an homologous way for the others.

Kurtosis and Skewness are measures that quantify aspects of the morphology of a

statistical distribution. Kurtosis measures the shape of the tails of a statistical distribution. Normal distributions have a value of Kurtosis of 3 and this value increases as the sparseness of the distribution increases. Skewness measure the symmetry around the mean value of a given probabilistic density function. It is zero when the distribution is symmetric, negative when the probability of a value being inferior to the mean value is greater than being higher and vice-versa. Both values are calculated regarding the concept of *central moment* which is calculated by Equation 3.6.

$$\mu_k = E[(X-\mu)^k] = \sum_{i=1}^{\infty}(x_i-\mu)^k P(X=x_i) \tag{3.6}$$

where X is a discrete variable, $P(X=x_i)$ the probability function and μ is the mean value of X. Skewness and Kurtosis are calculated by Equation 3.7, by β_1 and β_2 respectively.

$$\beta_1 = \frac{\mu_3}{\sigma^3}; \qquad \beta_2 = \frac{\mu_4}{\sigma^4} \tag{3.7}$$

For samples of n values, skewness and kurtosis may be estimated by Equation 3.8, by b_1 and b_2, respectively.

$$b_1 = \frac{n\sqrt{n-1}}{n-2}\frac{\sum_{i=1}^{n}(x_i-\bar{x})^3}{(\sum_{i=1}^{n}(x_i-\bar{x})^2)^{\frac{3}{2}}} \qquad b_2 = \frac{n(n+1)(n-1)}{(n-2)(n-3)}\frac{\sum_{i=1}^{n}(x_i-\bar{x})^4}{(\sum_{i=1}^{n}(x_i-\bar{x})^2)^2} \tag{3.8}$$

Linear regression consists on the approximation of the analysed instance by a straight line using a function $y = mx + b$, in which y is the resulting line parametrised by m, the slope, and b, the y-intercept point. These parameters are used to represent the instance.

Zero crossing rate corresponds to the number of times that the analysed instance crosses the zero value, which corresponds to the number of times that its signal changes. The number of peaks is calculated in an analogous way but in terms of the first derivative, with the consideration that the signal of the derivative must variate, from positive to negative or vice-versa.

Moreover, polarity is the absolute value of the division between the maximum and minimum values of a given set, in this case, a segment.

The cumulative summation consists on representing each temporal position by the sum of the previous data points of the signal. This may capture fluctuations on the behaviour of the signal that is not easily observable on the original representation.

Lastly, the histogram corresponds to the summation of amplitude intervals, denominated as bins, which may capture the underlying statistic distribution that generates the observed signal.

3.3.2 Representation Transforms

Representation transforms attempt to capture characteristics in different domains or representations which may be difficult to perceive in the original time series.

Fourier transform is a widely known transform which changes the domain of a signal

that is a function of time into a signal that is a function of frequency. This transform is performed using Equation 3.9.

$$\hat{x}(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-i(2\pi f)t} \cdot dt \tag{3.9}$$

Fourier transform allows to analyse the frequency details by discarding all the information in the time domain.

Wavelet transform is a technique which allows to describe a finite energy signal into an approximation of the original signal plus a set of details that range from coarse to fine. Thus, it is possible to have a view of the principal trend of the original signal from its approximation and quantify localised changes in the details given by the transform, which permits to capture local structures. This transform has good localisation both in time and in frequency by the dilation and translation of the wavelet function through the original signal, and may be viewed as the cross-correlation between the wavelet and the signal at each scale and each time instant. The referred wavelet, $\psi(t)$, is a localised structure and it can be selected from a wide range of known families, such as the Mexican Hat, Daubechies or Haar, it might be a portion of the signal or every function that respects the following conditions:

1. The energy of the considered function must be finite, this is: $E = \int_{-\infty}^{+\infty} \left|\psi(t)\right|^2 \cdot dt \neq \infty$.

2. Given the Fourier Transform of the wavelet function, $\hat{\psi}(f)$ (Equation 3.9), the admissibility constant is given by Equation 3.10.

$$C_g = \int_0^{+\infty} \frac{\left|\hat{\psi}(f)\right|^2}{f} \cdot df \tag{3.10}$$

The condition is that C_g is finite, which means that the wavelet function has zero mean.

3. For wavelets in the complex domain, Fourier Transform of the wavelet function must be real and the negative part of the transform must be zero.

Wavelet transform is performed by means of the wavelet function, that may be dilated and translated by parameters a and b respectively, according to Equation 3.11, where $w(a)$ is a weighting function dependent of the translation parameter and ψ^* denotes the conjugate of the wavelet function. In this dissertation, it was used the package provided in [40].

$$T(a,b) = w(a) \cdot \int_{-\infty}^{+\infty} x(t) \cdot \psi^*\left(\frac{t-b}{a}\right) \cdot dt \tag{3.11}$$

Principal Component Analysis (PCA) transforms the original data into the most relevant dimensions, which correspond to the dimensions where the variance is maximum and the correlation between variables is minimum. In order to apply this transform on a

time series it is necessary that each time instant to be considered a different dimension of a vector. In an application with various segments it is possible to perform the transform, because the correlation and variance can be calculated considering each segment as a different vector. This transform is further described in Section 3.3.4.

Independent Component Analysis (ICA) is a special case of blind source separation, which is the task of separating various sources that generate a set of mixed signals. Thus, an observation is characterised by Equation 3.12, where $j \in \{1, 2, ..., n\}$, n is the number of sources and, in this case, the number of observations, $\mathbf{s}$ are the sources and $\mathbf{A}$ is the matrix that models the observed components.

$$x_j = a_{j1}s_1 + a_{jn}s_n + ... + a_{jn}s_n \Leftrightarrow \mathbf{x} = \mathbf{As} \tag{3.12}$$

PAA in the amplitude domain (AD-PAA) was proposed in [41] and consists on the division of the amplitude domain in equal-length intervals, such as the histogram. Then each interval is represented by its mean value, as in regular PAA.

PAPR segments a time series in the amplitude domain such that each interval has the same number of points. Then, each corresponding interval is represented by a set of statistical features.

Subsegment analysis is analogous to PAA, but each segment is re-segmented and each subsegment is represented by a set of statistical features, namely, skewness, kurtosis, range (maximum minus the minimum value), polarity, IQR, standard deviation, mean and linear regression.

3.3.3 Comparison Metrics

3.3.3.1 Euclidean Distance

Euclidean distance measures the distance point by point between two time series by Equation 3.13:

$$d(x, y) = \sqrt{\sum_{i=0}^{N} (x_i - y_i)^2} \tag{3.13}$$

Considering X and Y the two time series and N the length of the time series. This means that the time series must have the same length and, once the distance is calculated point by point, similar signals that are out-of-phase or that have time distortions will have a large distance to each other. Furthermore, in order to compare signals with different lengths it is required to interpolate the signals, which may be a source of error.

Euclidean distance has three properties: **(1)** $d(x, y) \geq 0$ and $d(x, y) = 0$ only if $x = y$; **(2)** it is symmetric, which means $d(x, y) = d(y, x)$; **(3)** it respects the triangle inequality: $d(x, y) \geq d(x, z) + d(y, z)$.

3.3.3.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is a comparison metric which solves the problem of out-of-phase and time warped time series that exists in the analysis using Euclidean Distance. To do that, it is constructed a cost matrix, **C**, of the comparison of two time series, which has small cost if the signals are similar or large if they are very distinct. The matrix is constructed as follows: the cell $C_{i,j}$ is the result of the cost function of the signals in i and j, respectively, $C_{i,j} = dist(x_i, y_j)$. Therefore, each line corresponds to the distance between a given point from the first time series and all points of the second time series. The matrix has the dimension of $M \times N$, where M is the length of X and N is the length of Y. Figure 3.5 shows an example of a cost matrix computed using DTW algorithm between two different time series.

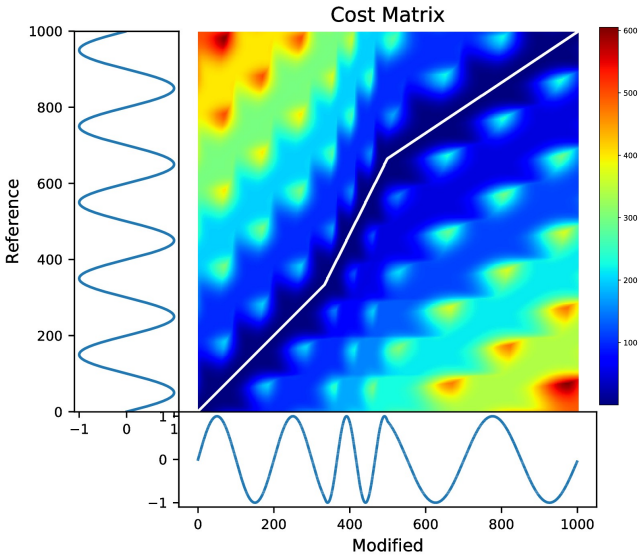

Figure 3.5: Cost matrix computed using DTW algorithm. The white line corresponds to a possible warping path.

Having constructed the matrix, the task is then to find the path W in which the overall cost, which is the sum of the cells $w_k = (i, j)$ that compose it, is minimised, corresponding to the optimal alignment. That path is shown by the white line in the cost matrix in Figure 3.5 and is constructed by:

- **Boundary condition** - the first point of the path must be $w_1 = (1, 1)$ and the last point must be $w_L = (M, N)$, where L is the length of the path.

- **Monotonicity condition** - the points that compose the warping path must be constantly monotonic, $i_1 \leq i_2 \leq ... \leq i_N$ and $j_1 \leq j_2 \leq ... \leq j_M$.

- **Step size condition** - there must be some advance in the construction of the warping path such that $(w_k - w_{k-1}) \in (0, 1), (1, 0), (1, 1)$.

Following these rules the possible paths are computed and the one with minimum total cost is selected as the optimal path, w^*. Then, the distance is given by:

$$d_{DTW}(X,Y) = \frac{1}{K}\sqrt{\sum_{k=1}^{K} w_k^*} \tag{3.14}$$

where K is the length of the optimal warping path and is used to achieve normalisation of the distance value. Figure 3.6 shows two time series warped in time and the optimum path allows the pointwise comparison represented by the black lines between the reference time series (green curve) and the warped time series (blue curve).

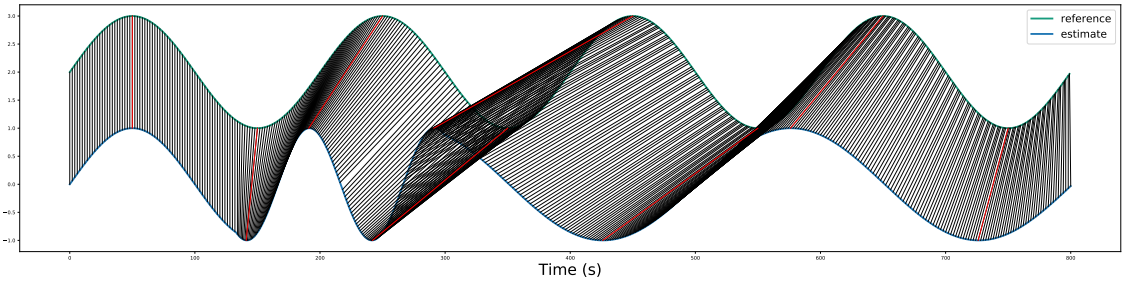

Figure 3.6: Optimum warping path found using DTW. The black lines correspond to the points that are compared given that path and the total distance is given by the normalised sum of the distance between each pair of compared points. The red lines are the reference lines, the green curve is the reference curve and the blue curve is the time warped curve.

3.3.3.3 Time Alignment Measurement

Although DTW distance allows the comparison of time series with different lengths that may be warped in time, it does not quantify the warping degree of the two compared time series. Thus, there may be cases in which the series are similar in shape, but have some distortion on the time domain that DTW is unable to identify. With the intent to provide information about how time warped are two time series, the authors of [42] proposed the Time Alignment Measurement (TAM).

Considering X and Y distinct time series of length N and M, respectively, a delay of Y in relation to X is represented by $\overrightarrow{\theta}_{XY}$, an advance of Y in relation to X is represented by $\overleftarrow{\theta}_{XY}$ and a phasing time is represented by $\overline{\theta}_{XY}$. These values are retrieved from the optimal path given by the DTW algorithm and calculated iteratively by Equation 3.15.

$$\begin{cases} \overrightarrow{\delta}_i = 1, & \text{if } (w_k - w_{k-1}) = (1,0) \\ \overleftarrow{\delta}_i = 1, & \text{if } (w_k - w_{k-1}) = (0,1) \\ \overline{\delta}_i = 1, & \text{if } (w_k - w_{k-1}) = (1,1) \end{cases} \implies \begin{cases} \overrightarrow{\theta}_{XY} = \sum_{i=1}^{k} \overrightarrow{\delta}_i \\ \overleftarrow{\theta}_{XY} = \sum_{i=1}^{k} \overleftarrow{\delta}_i \\ \overline{\theta}_{XY} = \sum_{i=1}^{k} \overline{\delta}_i \end{cases} \tag{3.15}$$

Then, it is defined the fraction of time in which there is an advance ($\overrightarrow{\psi}_{XY}$), a delay

($\overleftarrow{\psi}_{XY}$) and in phase ($\overline{\psi}_{XY}$) and the TAM value ($\Gamma_{XY}$) is calculated with resource to those:

$$\overrightarrow{\psi}_{XY} = \frac{\overrightarrow{\theta}_{XY}}{N}; \qquad \overleftarrow{\psi}_{XY} = \frac{\overleftarrow{\theta}_{XY}}{M}; \qquad \overline{\psi}_{XY} = \frac{\overline{\theta}_{XY}}{min(N,M)} \tag{3.16}$$

$$\Gamma_{XY} = \overrightarrow{\psi}_{XY} + \overleftarrow{\psi}_{XY} + (1 - \overline{\psi}_{XY}) \tag{3.17}$$

TAM value is restricted to values between 0, corresponding to the value with no dephasing, and 3, which is the value corresponding to a total dephasing.

Figure 3.7 shows the comparison between four time series to a reference, in which the compared time series are warped in time, compared to the reference. DTW distance is unable to distinguish the time series, while TAM value distinguishes them.

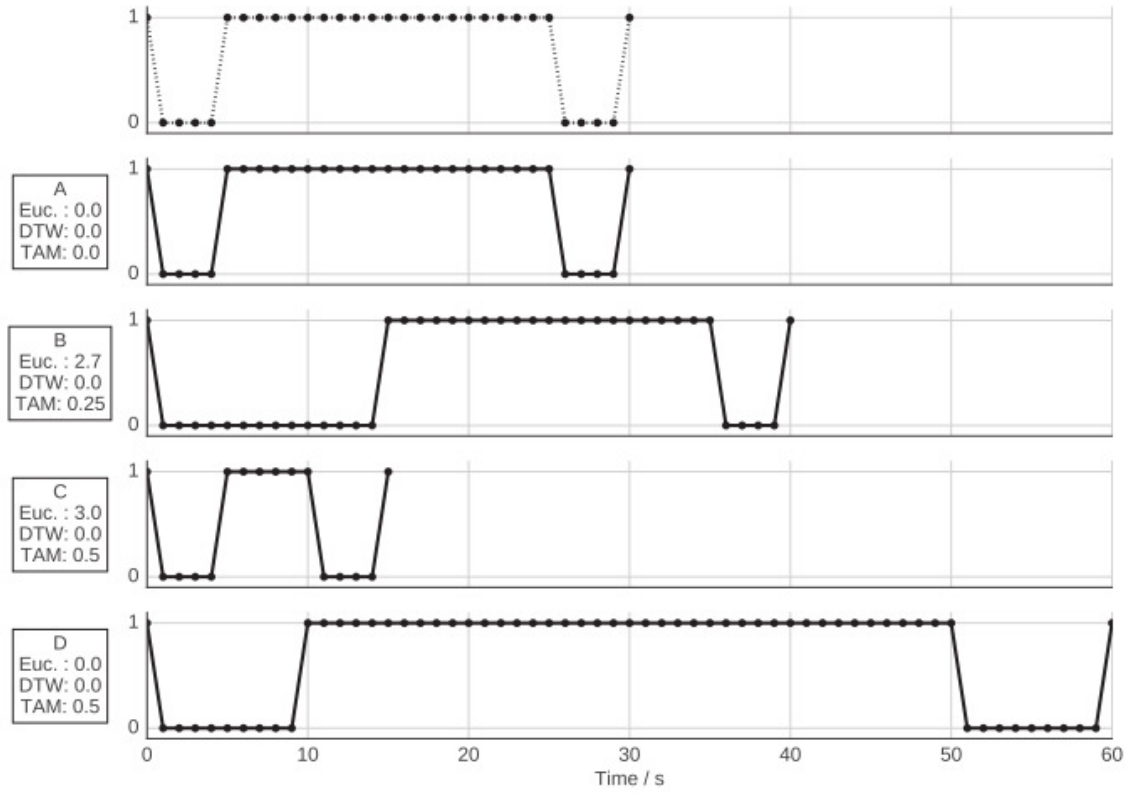

Figure 3.7: Time warping influence on the TAM value. The signal represented by dashed lines is the reference. Signal A is exactly the same, and so, all distances have 0 value. Signal B has a delay at the beginning and the TAM value reflects it. Signal C represents a compression of the original signal by half and Signal D representation a dilation of the signal by double, and so the TAM value is able to reflect it, though it does not distinguish the compression from the dilation (this could be achieved by calculating $\overrightarrow{\psi}_{XY}$ and $\overleftarrow{\psi}_{XY}$). Note that DTW is unable to distinguish all signals. From [42].

3.3.3.4 Pearson's correlation coefficient

Pearson's correlation coefficient gives the value of correlation between two time series of the same length, which is the similarity of their trends and is calculated using Equation

3.18:

$$\rho_{XY} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \quad \rho_{XY} \in [-1,1] \tag{3.18}$$

where $cov(X,Y)$ is the value of covariance between X and Y and σ_X, σ_Y are respectively the standard deviation of X and Y. ρ_{XY} takes the value of 1 in case X and Y are highly correlated, which means that X increases when Y increases, takes the value of -1 when they are inversely correlated, meaning that an increase of X is accompanied by a decrease of Y, and 0 means that there are no correlation and the behaviour of X can not be inferred by the behaviour of Y.

3.3.3.5 Cosine Similarity

This metric measures the cosine of the angle between two vectors. Thus, it provides information about the relative orientation between vectors, in detriment of the information about the magnitude of those vectors. It is bounded by -1 and 1. It takes the value of 1 when the vectors are oriented at 0º, 0 when the angle between them is 90º and is -1 when they are diametrically opposed. The equation which gives the similarity is:

$$similarity = cos(\theta_{xy}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \times \|\mathbf{y}\|} = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^2 \times \sum_{i=1}^{N} y_i^2} \tag{3.19}$$

In our case, the segments of the time series are treated as vectors with dimension equal to their length in order to compute the cosine similarity value.

3.3.3.6 Comparison setup

In this work, there are two settings in which these metrics were applied. The first refers to the comparison between each segment to all the others, such as shown in Figure 3.9. In this setting each segment is represented by the mean value of the comparison metric to all others.

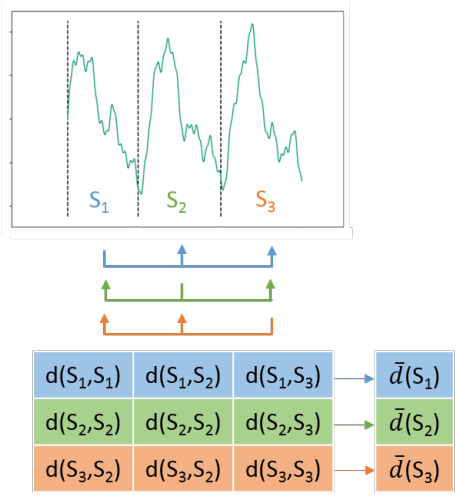

Figure 3.8: Comparison between all segments of a time series.

The other settings in which distance metrics may work is in comparison with segment models, in which each segment is represented by the distance to the model. Namely, in this work, these models may be constructed by three different processes, as represented in Figure 3.9, which are:

- **Mean Wave** - each segment is extrapolated to the length of the longest one and the mean value of each position is calculated, forming the mean wave of the signal;
- **Best Waves** - by the computation of the distance matrix using the Euclidean Distance between every segment, which are previously extrapolated to the length of the longest segment, the n segments with the minimum distances that are not 0 are chosen (every segment has a value of 0 in the array of distances, which corresponds to the distance to itself). These are considered the best waves, and the mean wave of those n segments is calculated;
- **Worst Waves** - analogously to the Best Waves, it is computed the distance matrix of all segments using the Euclidean Distance, but here the chosen waves are the ones that have a higher distance, which are considered the worst waves. Then the mean wave of the selected is computed to serve as model.

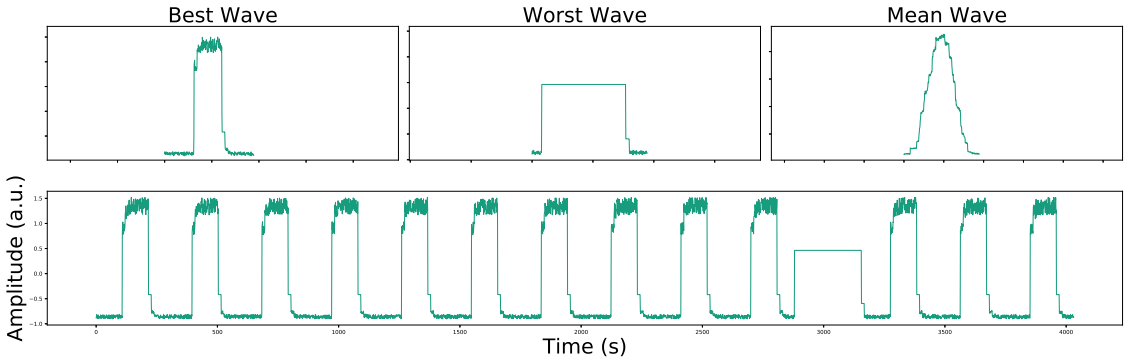

Figure 3.9: Illustration of all settings for time series comparison. Bottom figure shows the original time series. Top-left image shows the considered best wave; top-center shows the considered worst wave; top-right figure shows the mean wave calculated from all segments.

3.3.4 Dimensionality reduction

Feature selection is performed manually by the user, but a final step is applied in order to transform the set of features to a low dimensional and yet relevant space. This step consists on the application of Principal Component Analysis (PCA) after a Z-score standardisation (see Section 2.5.1).

PCA is centered on the idea of dimensionality reduction while retaining as much variation as possible present in the data set [43]. Dimensionality reduction is achieved by computing the Principal Components (PCs), which are new sets of variables that are

uncorrelated and are ordered so that the first few have most of the information of all of the original variables.

Computation of PCA starts by assuming that a vector $\mathbf{x}$ is composed by p random variables and the structure of the covariances or correlations between the p variables are of interest. The computation of PCs starts by looking for a linear function $\alpha_1^T\mathbf{x}$ having maximum variance such that:

$$\alpha_1^T\mathbf{x} = \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1p}x_p = \sum_{i=1}^{p} \alpha_{1i}x_i \tag{3.20}$$

The next step consists on finding a second linear function $\alpha_2^T\mathbf{x}$ uncorrelated with $\alpha_1^T\mathbf{x}$ with maximum variance. This process can be performed until finding p PCs, always with the constraint that the k^{th} Principal Component, $\alpha_k^T\mathbf{x}$, has maximum variance while being uncorrelated to all the previous PCs, $\alpha_1^T\mathbf{x}, \alpha_2^T\mathbf{x}, ..., \alpha_{k-1}^T\mathbf{x}$. To calculate the PCs it is required the consideration of the covariance matrix Σ. Accordingly to [43], the k^{th} PC is $z_k = \alpha_k^T\mathbf{x}$ where α_k is the eigenvector of Σ correspondent to the k^{th} largest eigenvalue, λ_k. Furthermore, if α is normalised such that $\alpha_k^T\alpha_k = 1$, then $var(z_k) = \lambda_k$, where var denotes variance. Hence, the eigenvalues are ordered by their value, which allows to obtain the PCs by decreasing order of variance, meaning that if, for example, the j^{th} PC is chosen, then it is certain that it is the PC with the j^{th} higher variance. Figure 3.10 shows an example of application of PCA in a simulated data set.

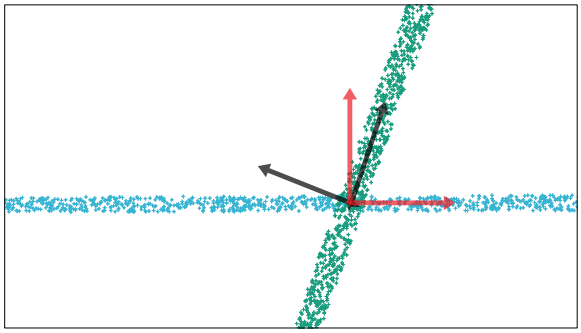

Figure 3.10: PCA representation concerning a simulated data set. Green points represent the original data set and the black arrows the corresponding PCs. Blue points represent the transformed data set maintaining the same number of dimensions, and the red arrows are the corresponding PCs.

The biggest problem of PCA is the fact that the obtained PCs lose the physical meaning that the original features have due to the space transformation.

PCA (algorithm from [44]) is applied with a variance threshold of 0,95 to the set of features, which allows to keep only the PCs with variance higher than the threshold. Thus, the curse of dimensionality can be averted, the performance of the framework is enhanced and the achieved results may be better than in the case in which the raw

features are introduced for the clustering algorithm (see Section 3.4). The transformed set of features will then be the input for the clustering algorithm.

3.4 Clustering Algorithms

Clustering algorithms are unsupervised mechanisms that attempt to find the best form to group data points. The most commonly used is k-means which starts by randomly producing k centroids in the data space. Then, the pairwise distance between each data point and each centroid is calculated and each data point is assigned to the closest centroid. After that, in an expectation maximisation fashion, the centroid position is recalculated to correspond to the arithmetic mean of the points that belong to its centroid. This process is executed iteratively until convergence. The major drawback of this algorithm is the fact that it needs the information of the number of clusters *a priori*. Furthermore, k-means assumes that every data point belongs to a cluster, which may not hold true for anomalous instances, depending on the domain of application. Additionally, due to the computation of the pairwise distance, it is assumed that data clusters form hyper-spheres, which can be too simplistic in many cases.

Due to these problems, it was chosen the *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). It is based on the concept of density instead of distance. In the scope of anomaly detection, the assumption is that normal instances form dense clusters while anomalous instances are sparser in space.

DBSCAN takes two hyper-parameters: ε and $minPts$ which define the concept of *dense samples*. There are three different types of data points according to this algorithm and to those parameters. The first type is the *core sample* data point, which has at least a number of $minPts$ at a distance inferior to ε, defined in [45] as neighbours. These data points represent the denser part of the clusters, because they have the highest number of points close (at a distance of ε or less) to them. Then there are the *non-core samples*, which are points that are within the range of ε of a core sample, but have not enough neighbours to be considered core points. These points form the fringes of the cluster and are also part of the cluster. Thus, a cluster is constructed by taking a core point, find other core points in its neighbourhood and continue until finding the fringes of the cluster. This means that there may exist points that do not belong to any cluster, because they are too sparse and are further than ε to any core sample, which are considered *outliers*. An illustration of these concepts is given in 3.11.

It is possible to perform an analysis about the influence of the parameters, which is essential to understand the proper functioning of DBSCAN. Starting with the increase of ε and keeping the value of $minPts$ constant, the range of search of neighbours increases, which allows the appearance of more neighbours, leading to a less sensitive application, because more points may be included in each cluster, which means that the clusters can be larger, possibly leading to a reduction of the number of clusters. On the opposite, the decrease of ε leads to a more sensitive solution because the instances must be nearer to

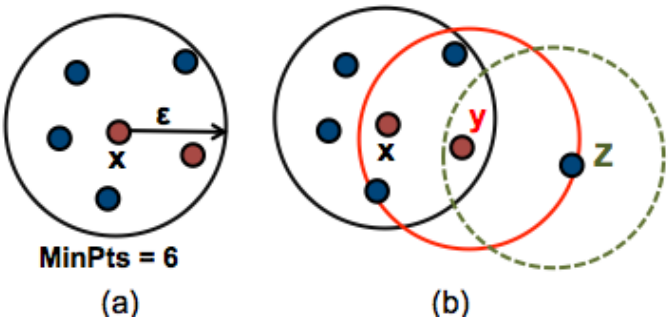

Figure 3.11: Illustration of the types of points considered in DBSCAN. On the left-pane it is presented the notion of neighbourhood, in which a data point has a number of $minPts$, counting with itself, within the range of ε. On the right, it is given an example of the three types of points: x is a core point, y is a non-core point and z is an outlier. Image source: [46].

each other in order to form clusters, leading to solutions with more outliers and more and smaller clusters. On the other hand, if the value of ε is kept constant and the variation is made on the number of $minPts$, the analysis is reciprocal. An increase of $minPts$ means that it is necessary to have more neighbours in order to a sample to be considered a core sample, which means that there is a higher sensitivity and the results are similar to a decrease of ε. A decrease of $minPts$ leads to a less sensitive outcome because the number of neighbours required for a sample to be considered a core sample is less, and so it is natural the appearance of a higher number of core samples, and the results are similar to the results in case of an increase of ε.

The appropriate selection of the parameters can be made manually, by empirically analysing the achieved results and adapting the parameters in accordance with those results, or it may be made automatically by the analysis of the k-Nearest Neighbour (k-NN) curve, in which k is the number $minPts$. The k-NN curve is constructed by calculating the Euclidean distance between each sample and all the others. Then it is calculated the mean value of the k lowest distance values, which means the mean distance to every sample to the k^{th} nearest neighbour, and those values are sorted in a descending order. An example is shown in Figure 3.12. Given the value of $minPts$, the estimation is now relative to ε. It was tested two ways to select the appropriate value that were based on the highest variation of the first derivative of the curve, corresponding to the value from which the variation of distances varies the most. Prior to that value there is a great variation of distances and after that value the distances vary very little. That value was estimated by two approaches: (1) assuming that it corresponds to the point where the second derivative of the curve is the highest, which is the point with highest inflexion or (2) it can be assumed that it corresponds to the value in which the first derivative is equal to −1. The developed framework accepts the manual choosing of ε or it can calculate it automatically by (2), which was the method that demonstrated to hold better results.

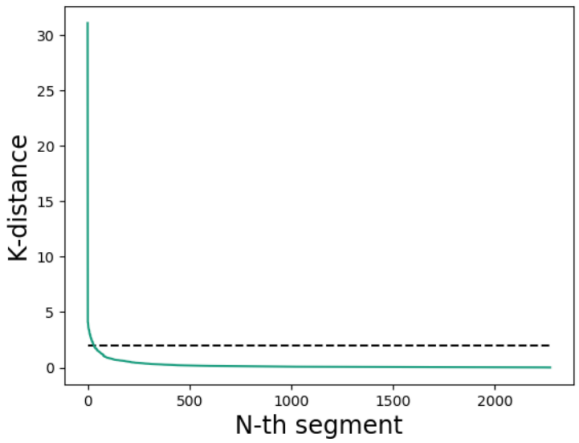

Figure 3.12: k-NN distance curve is constructed by calculating the mean distance to the k^{th} nearest neighbours for every sample, which represents a segment in our case, and sorting those values in an ascending order. The dashed line corresponds to the value of ε calculated by (2).

The developed framework gives a binary classification based on the clustering outcome. DBSCAN is deterministic in the sense that the formed clusters are always the same given the same parameters as input, but the index attributed to each cluster can vary with the order of the input. Therefore, in this application, the normal label is assigned to the instances that populate the most populated cluster, and the others are labelled as anomalous. This is important in cases in which anomalies are similar to each other, such as classes of arrhythmia in ECG signals, which may form clusters of anomalous instances.

Furthermore, in the scope of this dissertation, it were not made any modifications to the original DBSCAN algorithm(from [44]), but there were some considerations that were made in order to respect the assumptions made initially, specifically in relation to the third assumption, which states that the number of anomalies is always less than the number of normal instances. Those considerations were, given that the input to the algorithm are the features extracted from each segment representing the samples, if the number of segments considered to be anomalous are higher than the number of segments classified as normal, the value of ε increases by 10% and the classification/clustering process is performed again. This process is repeated until the number of normal instances is higher than the number of anomalous instances.

The main advantages of DBSCAN in relation to k-means is that it does not require the number of clusters *a priori* and it allows the presence of outliers, which correspond to sparse data points. In fact, it is able to find various clusters based on the input parameters defined by the user. Furthermore, contrary to k-means, it does not assume the shape of the data points. Nevertheless, the fact that it needs ε and $minPts$ is a major drawback of this algorithm because it is markedly influenced by those parameters in order to compute meaningful clusters.

3.5 Summary

In this chapter it was presented an extended view about the developed framework. It was described a new unsupervised segmentation algorithm that is able to extract every cycle of a repetitive signal originated by a cyclic event. Then, the extracted cycles pass through a features extraction phase which helps to represent them in a concise way. The extracted features, which represent each cycle, are transformed by the application of the PCA algorithm and then presented as the inputs to a clustering algorithm, DBSCAN, based on the concept of density to distinguish the samples. Several considerations are then made on the results in order to assign each cycle to a class, that may be normal or anomalous.

The main advantage of this framework is the fact that it may be applied in a wide range of applications and it functions in an unsupervised fashion, which dispenses the human work of labelling every instance for training the classification model. Nevertheless, once the framework may be applied to general cyclic time series, it might not be expected a performance similar to the obtained with specialised approaches.

Chapter 4

Experimental Evaluation

4.1 Datasets Description

Given the developed framework, it is important to test its application in various domains in order to demonstrate the potential to generalise over a wide range of domains. Thus, there were used four different datasets from different domains. The first two datasets were composed of synthetic signals while the last two are real world signals. In this chapter, these datasets will be further described and the respective results will be presented.

4.1.1 Numenta Anomaly Benchmark

Numenta is a project based on the research about the biology of the neocortex in order to create complex technology and machine intelligence. This technology is based on the concept of Hierarchical Temporal Memory (HTM), which is a biological-constrained theory of intelligence [47] that attempts to integrate the available information about the neocortex of the human brain in order to create intelligence. The developed algorithms have the ability to learn from generic time series and detect anomalies in various domains in real-time on streaming data, rather than in batches.

In order to test the algorithm developed by Numenta, it was created a public access dataset containing real world and synthetic data which contains time series of exclusively normal instances and also time series that contain anomalies mixed with normal instances. The data originated from real world is not cyclic and, thus, can not be used to test the framework developed in the context of this dissertation. On the other hand, there are artificial signals by Numenta that are periodic and are divided in cyclic with no anomalies and cyclic with anomalies. The availability of this dataset allows the comparison of other methods to the HTM constrained system and is also an important benchmark tool for

anomaly detection algorithms.

Aiming for an objective comparison of results obtained with different algorithms, it is proposed a new scoring method, which takes into consideration the delay or the advance of the detected anomalies in relation to the actual anomalies. It is stated in [48], where the new scoring was proposed and the dataset was presented, that the evaluation with standard metrics do not account for early detection and thus, are not suited to analyse anomaly detection techniques. Yet the new scoring is only intended to be applied in real time applications and in which the anomalies are detected in an instant in time rather than in a segment and, so, it is not adequate to be used for scoring the results obtained by the framework proposed in this dissertation.

The selected artificial signals are composed of 4.032 data points each and are in total 9 signals, thus, making 36.288 data points analysed, with cycles of different morphologies and noise in order to mimic real world repetitive data (Figure 4.1). There are two other signals that are included in the artificial signals provided by NAB, but once they are not cyclic, they will not be analysed.

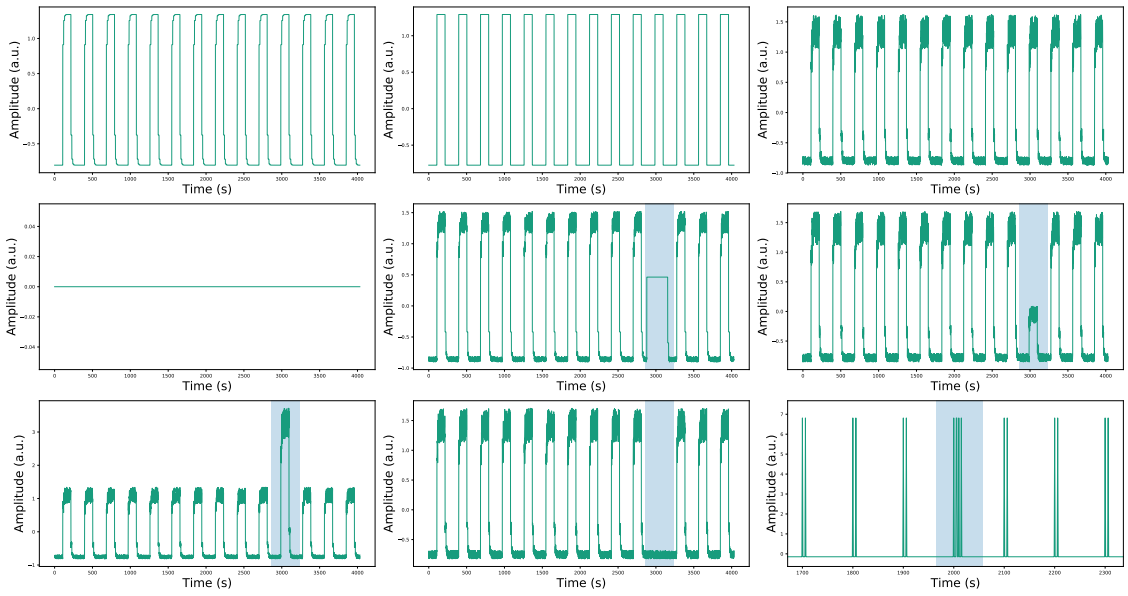

Figure 4.1: Signals selected from the Numenta Anomaly Benchmark. The existing anomalies are identified by blue shades.

4.1.2 Pseudo Periodic Synthetic Time Series

The Pseudo Periodic Synthetic Time Series dataset is a dataset provided by the Center for Machine Learning and Intelligent Systems of the Bren School of Information and Computer Science of the University of California and is publicly available[1]. It is composed of 10 artificial signals composed of 100.001 data points each, making 1.000.010 data

[1] available in https://archive.ics.uci.edu/ml/datasets/Pseudo+Periodic+Synthetic+Time+Series

points in total. These signals are repetitive, but the cycles are not exactly alike. They were produced by the Equation 4.1.

$$\bar{y} = \sum_{i=3}^{7} \frac{1}{2^i} sin(2\pi(2^{2+i} + rand(2^i))\bar{t}); \quad 0 \leq \bar{t} \leq 1 \tag{4.1}$$

where $rand(2^i)$ is a random integer between 0 and 2^i. Hence, the non exact repetitiveness is achieved by the introduction of the random parcel in the equation.

These facts make this dataset suitable for testing the framework proposed on this dissertation, but there is a crucial aspect lacking to these signals, which is the presence of anomalies. Thus, it was developed a new Anomaly Introduction Framework in order to introduce different types of anomalies in univariate time series in a controlled fashion, which allows the benchmarking of anomaly detection algorithms.

4.1.2.1 Anomaly Introduction Framework

During the course of this work, it was identified a lack of cyclic anomalous signals in literature or frameworks that would be able to generate them. In order to overcome this limitation, it was developed a new framework for anomaly introduction on univariate time series. With this framework, it is possible to choose the type of anomaly that is intended to be applied, the instants or temporal intervals in which it is to be applied and control the anomaly degree.

There are seven types of anomalies identified that can be introduced: three in the temporal domain and four in the amplitude domain. First, the temporal domain anomalies will be presented and in further analysis only the parameters that control the anomaly degree will be described (the others are the name of the anomaly and the instants or segments in which it is applied):

- **Linear distortion** - the selected segment is interpolated to the length assigned by the user. There is one parameter that has to be chosen, which is the degree of the distortion by $d \times length(segment)$, being d the chosen value. The resulting time vector is linear.

- **Non-linear distortion** - the time vector is modified by the application of the cumulative sum of Equation 4.2, leading to a monotonically crescent time vector that is not linear. Then, the considered segment is interpolated to the new time vector.

$$f(t) = \begin{cases} 0, & 0 < t < t_1 \\ \left| \frac{2t}{N-1} - 1 \right|, & t_1 \leq t < t_2 \\ 0, & t_2 \leq t \end{cases} \tag{4.2}$$

  where t_1 and t_2 are predefined and N is the length of the time series, therefore, there are no parameters to define.

• **Mirror** - the time vector of the chosen segment is inverted. Thus, the originally last point results on the first and so on. There are no parameters to select.

In Figure 4.2 it is shown the three types of temporal anomalies compared to the original signal.

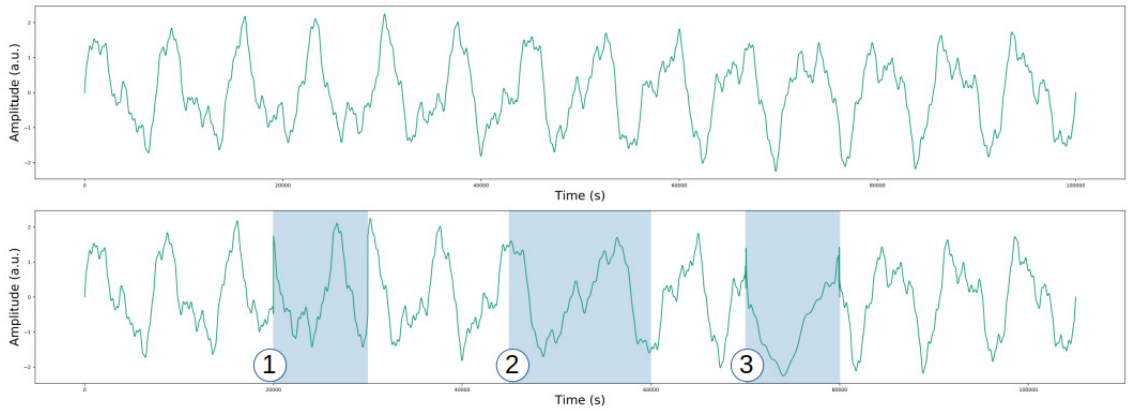

Figure 4.2: Anomalies in the time domain. At top, it is presented the original time series and at the bottom there are the three anomalies identified by the blue shades. 1. *mirror*; 2. *linear distortion*; 3. *non-linear distortion*.

Amplitude domain anomalies correspond to the alteration of the amplitude vector. The anomalies that might be modelled are:

• **Abrupt Single** - a value is added to a single data point. Though this type of anomaly is not considered in the proposed anomaly detection framework, it is a well known anomaly that exists in various types of data and, thus, it is important to model it. This anomaly requires only the value to add to the previously chosen instant in time.

• **Abrupt Segment** - a single value is added to a segment of the signal, producing a shift of that segment. Analogously to the case of Abrupt Single, it is required the value to add to the segment composed of the previously chosen instants.

• **Noise** - the amplitude vector is affected by white noise generated by a function which generates a number of values equal to the length of the segment in which the anomaly is to be applied. The values generated are random and follow a normal distribution with zero mean and standard deviation of one. The parameter that the user chooses is a value that is multiplied to the random array, which will signify the standard deviation of that array.

• **Smooth** - it is constructed a new amplitude vector by the sum of the original vector, S, and a Gaussian noise vector which is modelled by a sine function such that $S(t) - S(t+1) = \delta \sin(rand)$, where $rand$ is the t^{th} term of the noise vector and δ

is the maximum (minimum) amplitude that it may reach and is specified by the user, which ensures the degree of variation between adjacent points. The vector is iteratively modified as depicted in Algorithm 1 the number, θ, of times the user specifies. Lastly, the final vector is scaled by two user specified parameters, α and β, that correspond to the upper and lower amplitude limits.

Algorithm 1 Pseudo-code used to apply a smooth anomaly in the amplitude domain of a given signal.

Input: original signal, δ, θ, α, β
Output: signal modified in the amplitude domain

$array := \sin(rand(length(\text{original signal}))) \times \delta$;
$c\_array :=$ cumsum$(array)$;
while i **do** in range$(2, \theta)$:
    $c\_array \leftarrow cumsum(array)$;
    if then $\max(c\_array) - \min(c\_array)) > 2\pi$:
        $$var\_array := \frac{(c\_array - \min(c\_array))}{2\pi \max(c\_array) + \min(c\_array)};$$
    end if
    $array \leftarrow \sin(var\_array) \times \delta$;
end while
return original signal $+ \dfrac{(var\_array - \min(var\_array)) \times (\alpha - \beta)}{(\max(var\_array) - \min(var\_array)) - \dfrac{(\alpha - \beta)}{2}}$

In Figure 4.3 it is shown three types of considered amplitude anomalies compared to the original signal.

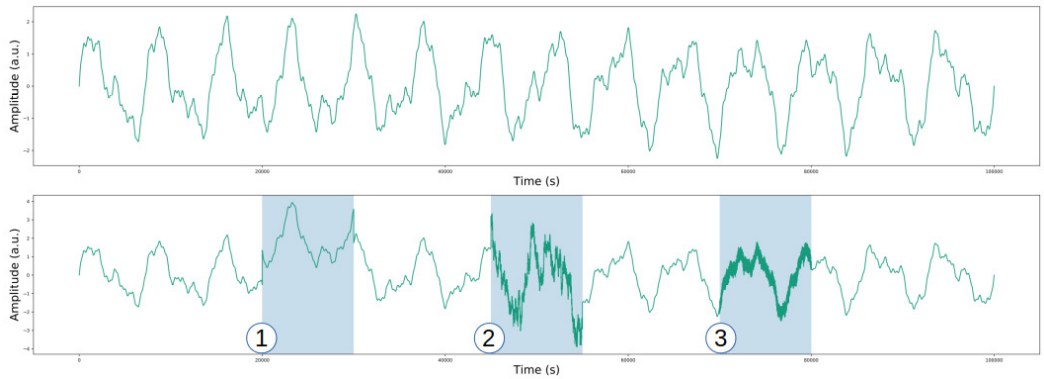

Figure 4.3: Anomalies in the amplitude domain. At top, it is presented the original time series and at the bottom there are the three anomalies identified by the blue shadows. 1. *abrupt segment*; 2. *smooth* anomaly; 3. *noise*.

Each anomaly may be applied using specific functions designed for the introduction of each type of anomaly. Furthermore, it is possible to produce a json file, such as the one in Figure 4.4, which is expected to have the presented structure, in which every anomaly is described and the application is made sequentially as the order in the file.

```
{Timestamps:
                {Type of Anomaly: ({'Time', 'Amplitude', 'Both'})},
        Parameters:
                    {(Type of Anomaly) Distortion Type: ({'linear', 'nonlinear'},
                    {'mirror', 'smooth', 'abrupt', 'noise'})
    (Type of Anomaly) Parameters:
        ({(int)}, {(list)})}}
@example:

{'[40000, 60000]':
                {'Type of Anomaly': 'Time',
    'Parameters':
                {'Time Distortion Type': 'linear', 'Time Parameter': 1.5}},

    '[12354, 14232]':
                {'Type of Anomaly': 'Amplitude',
    'Parameters':
                {'Amplitude Distortion Type': 'smooth',
    'Amplitude Parameters': [ 2, 3, [4,2]]}}}
```

Figure 4.4: Example usage of a .json file in order to introduce anomalies using the anomaly introduction framework. The first anomaly, corresponding to a time domain anomaly, specifically, a *linear distortion* with parameter of distortion of 1.5, is to be inserted from index 40.000 to 60.000. The second anomaly corresponds to the *smooth* anomaly in the amplitude domain which is to be inserted from index 12.354 to 14.232 with parameters $\delta = 2$, $\theta = 3$, $\beta = 4$ and $\alpha = 2$.

Thus, it is possible to, given a normal time series, produce a time series containing anomalous instances. Moreover, the instants in which the anomalies are introduced are registered, such as the code associated to each anomaly, allowing for multiclass classification of each type of anomaly, which is beyond the scope of this dissertation.

With this framework there were originated 500 signals from the first 10, with anomalies placed randomly, and chosen randomly (the Abrupt Single anomaly was not taken into consideration because it corresponds to a point anomaly). The anomalies occupied approximately 10% of each time series in consecutive instants and the parameters for each type of anomaly were determined randomly within ranges that were acceptable to be considered anomalous but not too different from normal signals.

4.1.3 MIT BIH arrhythmia database

MIT BIH arrhythmia database is a dataset composed of real world electrophysiological signals [49]. The signals are composed of data obtained from ECG recordings acquired in ambulatory. Electrocardiography signals represent the measurement of the electrical pulse that propagates through the cardiac muscle in order to stimulate it resulting on its normal behaviour, which enables the entry and exit of blood to and from the heart.

As depicted in Figure 4.5, normal ECG signals are essentially composed by five waves. The P wave is generated by atrial depolarisation, which leads to the atrial contraction. The QRS complex follows and represents the ventricular depolarisation leading to ventricular contraction. T wave represents the recovering of ventricular contraction and is labelled as a repolarisation wave. The correspondent wave with respect to atrial repolarisation seldom appears because of its low amplitude and the fact that the time required for repolarisation of the atria coincides with the appearance of the larger QRS complex.

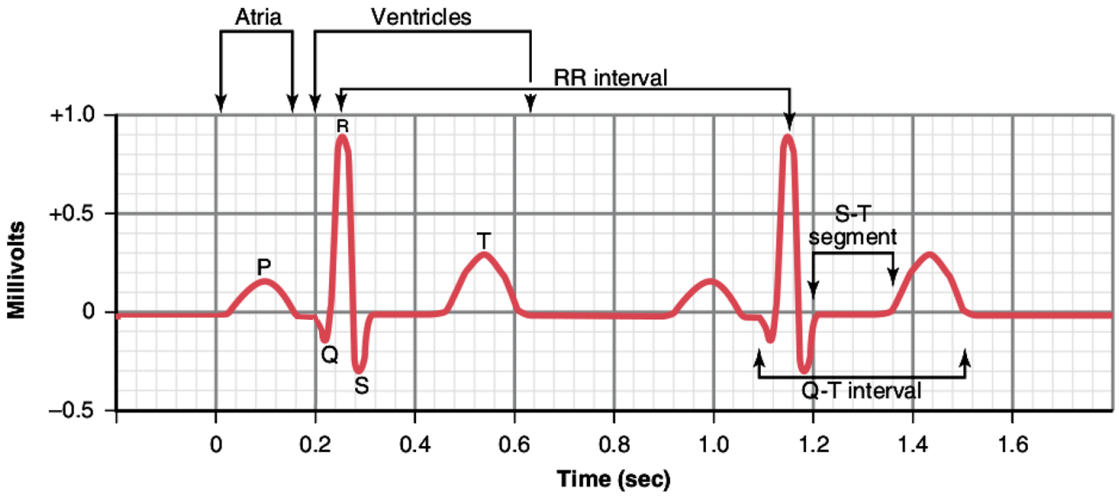

Figure 4.5: Normal ECG representation. Adapted from: [50]

The dataset is composed of 48 excerpts of two-channel half-hour ECG recordings of 47 subjects. From those records, 23 were randomly chosen from a set of 4.000 24-hour ambulatory ECG records collected from inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital. The 25 remaining signals were selected from the same set but in order to include less common, but yet clinically relevant arrhythmias.

The signals were acquired with a sampling frequency of 360 Hz and had a resolution of 11 bits over a range of 10 mV, composing a total of 62.400.000 data points. The dataset was annotated by two specialists in order to identify both the peak of the R waves and the diagnostic for the corresponding heartbeat. The dataset is composed of approximately 110.000 annotations, hence, 110.000 heartbeats.

This dataset has various types of arrhythmias that, accordingly to the Association for the Advancement of Medical Instrumentation (AAMI) may be grouped as in Table 4.1 [51]. Arrhythmia is a result of a malfunctioning of the heart reflected as an abnormality of its rhythm that may be provoked by [50]: (1) an abnormal rhythmicity of the pacemaker, which is the structure that starts the propagation of the electric potential through the heart and controls its rhythm; (2) a shift of the localisation of the pacemaker; (3) a block that may occur along the propagation of the electric pulse; (4) abnormal pathway of the propagation of the pulse through the heart; (5) spontaneous production of spurious electrical stimulus.

The 4 anomalous classes proposed by AAMI are depicted in Figure 4.6. Normal beat

Table 4.1: Table of the conversion of MIT BIH to AAMI classes. Based on [51].

AAMI class	MIT BIH class
Normal Beat (N)	Normal Beat (N) Left bundle branch block heart (L) Right bundle branch block heart (R) Atrial Escape Beat (a) Nodal (junctional) escape beat (j)
Supraventricular ectopic beat (S)	Atrial premature beat (A) Aberrated atrial premature beat (a) Nodal (junctional) premature beat Supraventricular premature beat (S)
Ventricular ectopic beat (V)	Premature ventricular contraction (V) Ventricular escape beat (E)
Fusion beat (F)	Fusion of ventricular and normal beat (F)
Unknown beat (Q)	Paced beat (/) Fusion of paced and normal beat (f) Unknown beat (Q)

corresponds to signals similar to Figure 4.5.

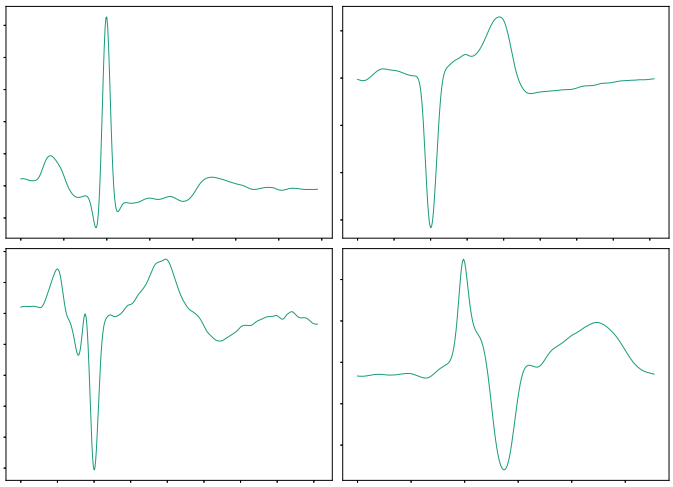

Figure 4.6: Arrhythmias in ECG signals. Top-left figure shows a *supraventricular ectopic beat*; Top-right figure shows a *ventricular ectopic beat*; Bottom-left figure shows a *fusion beat*; Bottom-right figure shows an *unknown beat*.

4.1.4 Human Motion on industrial scenario

Human Motion on industrial scenario (HMIS) dataset was acquired in a real world industrial environment with resource to an IoTiP (Internet of Things in Package) developed at Fraunhofer Portugal, that was integrated in a bracelet and strategically placed on the employees' dominant upper member. This placement allows to monitor the wrist's

movement performed by each monitored employee, which is relevant once the tasks performed involve predominantly upper member movements. Furthermore, it was made clear that the placement of the IoTiP was not in any way interfering with the work that was being performed, hence, leading to the conclusion that the solution may be used in an ubiquitous and unobtrusive way.

The IoTiP is a sensing device that integrates an Inertial Measurement Unit (IMU), which measures inertial data with resource to three sensors: an accelerometer, a gyroscope and a magnetometer [30]:

- **Accelerometer** - accelerometers are devices that measure the proper acceleration, which is the acceleration relative to free-fall and is the acceleration experienced by people and objects relatively to an inertial frame. Accelerometers measure both static and dynamic acceleration (for instance, gravity and human movement respectively) and those measurements may be represented in International System (SI) units (m/s^2) or in g-force units. Therefore, a static accelerometer measures an acceleration of about 1 g $\approx 9,81 m/s^2$ at rest, while a dynamic accelerometer measures 0 g = 0 m/s^2. The functioning of the accelerometer relies on the relative movement of a proof mass, that is also called seismic mass, attached to a mechanical suspension system with respect to a reference inertial frame. The mass moves relative to the system and that movement is measured through an electrical system, which commonly includes piezoelectric, that changes its electric potential difference by the change of its volume, piezoresistive, analogously to piezoelectric materials, changes its resistive properties with a variation of its volume, or capacitive components, which, with the variation of the distance of two plaques changes the capacitive properties, that is able to measure the proper acceleration of the proof mass.

- **Gyroscope** - gyroscopes measure both orientation and angular velocity. Traditional gyroscopes consist in a rotating disk, also called a rotor, attached to two gimbals and an axis about which it rotates. Due to conservation of the angular momentum, when the rotor is rotating, it tends to be rotating unless a force is applied. When a force with different direction of the rotation axis is applied, it produces a torque that induces precession. The measurement of the precession angle and velocity indicates the angular velocity and orientation of the source of the torque. There are also optical gyroscopes and vibrating mass gyroscopes. Optical gyroscopes consist on a laser which is split in two opposite directions inside a disk. If there is no rotation, both the laser beams arrive at the same time at a photoelectric sensor placed on the opposite site of the disk, but if the system rotates, the laser beam which travels contrary to the rotation, arrives earlier in relation to the beam that follows the rotation due to the Coriolis effect, which allows to calculate the angular velocity by the interval between the light beams. Vibrating mass gyroscopes are the most utilised in human motion monitoring due to their small size and low power consumption. Also based on the Coriolis effect, a resonator is placed on a rotating

disk and when a change in the angular velocity is applied to the disk, it is produced an oscillation orthogonal to the original vibrating direction. The new vibration can be measured and the angular velocity can be calculated by Equation 4.3.

$$F_C = -2m(\omega \times v) \tag{4.3}$$

where F_C is the Coriolis force, m is the mass of the rotating mass, v is the instantaneous velocity of the rotating mass relative to the moving object to which it is attached and ω is the angular velocity of the object.

- **Magnetometer** - magnetometers measure the direction, norm and variations of the magnetic field in a point in space and also the magnetisation of a material that exhibits magnetic properties. In smartphone applications, the most utilised magnetometers use the Hall effect to perform the measurements. Hall effect is the voltage difference that is created by the presence of an orthogonal magnetic field relative to an electric current. The presence of that magnetic field produces the deflection of the charges in a perpendicular direction to both the magnetic field direction and the current. That deflection is caused by the Lorentz force, which is created by the interaction of current and magnetic field, following Equation 4.4.

$$\overrightarrow{F_L} = q\overrightarrow{E} + q\overrightarrow{v} \times \overrightarrow{B} \tag{4.4}$$

where $\overrightarrow{F_L}$ is the Lorentz force, q is the electric charge, $\overrightarrow{E}$ is the electric field, $\overrightarrow{v}$ is the velocity of the electric charge and $\overrightarrow{B}$ is the magnetic field. Thus, measuring the voltage potential on the extremes of Hall elements it is possible to measure the magnetic field properties.

The combination of the three sensors allows a full comprehensive analysis regarding the movement of the monitored employee. The magnetometer is an important tool to assess the relative position of that employee because it may suffer high variations relative to the environment where it is inserted as demonstrated in [30].

The IoTiP is connected via Bluetooth to a smartphone app, where the gathered data is stored. With this app, it is possible to make various types of annotations and so, each acquisition was annotated in this application in the beginning of each new cycle and in every occurrence of anomalies, allowing to build the ground-truth segmentation and classification.

The acquisitions were made with the consent of 4 different workers at 3 different workstations where they were producing automobile parts. All tasks monitored were repetitive which made them suitable to be tested with the developed anomaly detection framework. The sampling frequency of the acquisitions were approximately 100 Hz and the total time of the acquisitions is around 4 hours and 20 minutes, thus making a total of 1.520.210 data points.

Furthermore, the considered anomalies consisted on unexpected breaks of the repetitive work that could be provoked by machine idle, defective pieces on the production line,

interruptions by co-workers, lack of pieces in the production line, a thorough analysis of the pieces that are being utilised to assess their quality, sign papers and machinery breakdown. Figure 4.7 shows an anomaly in this context, identified by the red shade, which consisted of an thorough analysis of an item which was wrongly assembled at the first try.

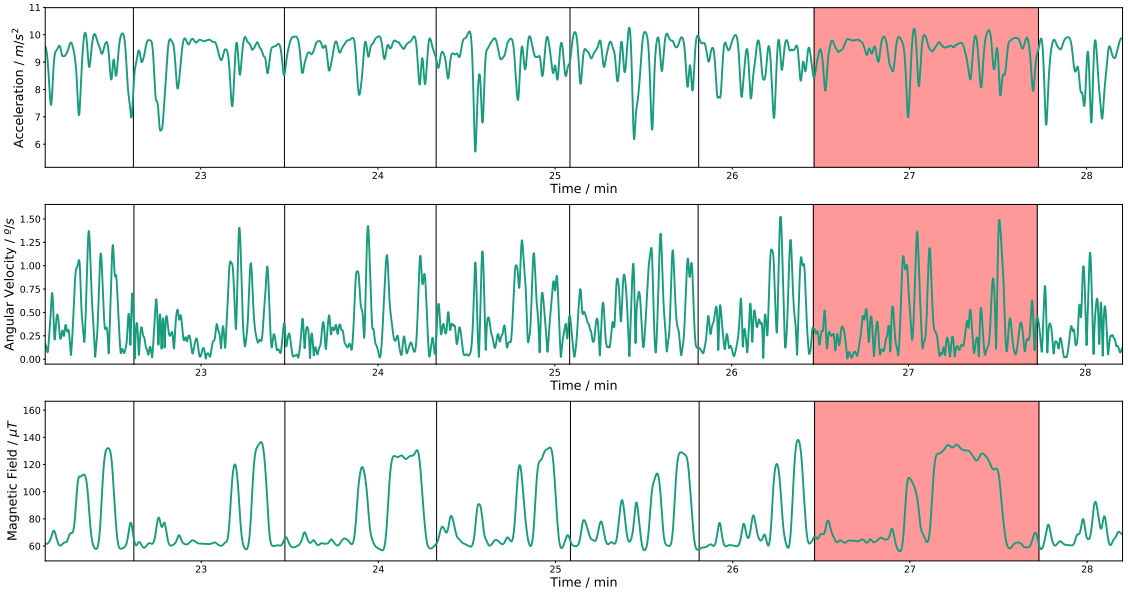

Figure 4.7: Anomaly in the HMIS dataset. Each plot corresponds to the magnitude of each sensor: top plot corresponds to the accelerometer; middle plot corresponds to the gyroscope; bottom plot corresponds to the magnetometer. The anomaly is idetified by the red shade in all plot and is more evident in the magnetometer data.

4.2 Results

In this section the results achieved with the proposed anomaly detection framework for each dataset will be presented and discussed. The considerations and pre-processing steps will also be described which vary for each dataset. In order to obtain the results, all datasets were previously annotated and a comparison was made between the obtained outcome of the proposed framework and the ground-truth labels.

4.2.1 Numenta Anomaly Benchmark

As described, NAB is composed of cyclic synthetic data that is ordered as to simulate quasi-periodic time series.

In the case of the proposed anomaly detection framework, the classification is made by segments which in theory correspond to cycles of periodic signals. Thus, in order to compare the annotations and the outcome of the proposed framework, it was considered that a true positive is each cycle assumed anomalous that contains a data point considered

as anomalous. A false positive is a cycle assumed as anomalous that does not contain any annotation of anomaly. The true negatives are the segments considered as normal that do not contain any annotation and, lastly, false negatives are the segments that are labelled as normal containing an annotation of anomaly.

The results and corresponding confusion matrix are presented on Tables 4.2 and 4.3. These results were obtained using the mean, maximum, minimum, median, inter-quartile range and skewness values followed by the procedures described in Chapter 3. The choice of the parameters to use in DBSCAN was performed empirically by observing the results and tuning the parameters in order to optimise the achieved results. This is not ideal, because in real life scenarios it is impossible to tune the parameters to new signals without prior knowledge about them. Nevertheless, the parameters are the same for all signals: $minPts = 5$; $\varepsilon = 5$. Furthermore, it was applied a pre-processing step prior to segmentation which consisted on the application of a mean filter. The indexes of segmentation were obtained with resource to the unsupervised algorithm proposed in Section 3.2 and applied on the original signal, and, thus, the segments are composed of the original signal.

Table 4.2: Confusion matrix referent about the Numenta Anomaly Benchmark dataset.

	True Label	
Predicted Label	Normal	Anomalous
Normal	141	0
Anomalous	1	5

Given the confusion matrix present in Table 4.2, it is possible to calculate the metrics that are usually used to assess the classification in classification problems. Those metrics are shown in Table 4.3.

Table 4.3: Results of anomaly detection using the Numenta Anomaly Benchmark.

Metric	Value (%)
Accuracy	99,3
Specificity	99,3
Sensitivity	100,0
Precision	83,3
F1 score	90,9
False Negative Rate	0,0
False Positive Rate	0,7
False Discovery Rate	16,7
Negative Predictive Value	100,0

These results are clearly indicative of overfitting due to the optimisation of the parameters that took into account all signals. Nevertheless, it is clear that the framework presents great potential to detect anomalies in time series.

To analyse the achieved results, the accuracy metric is not the ideal in the case of anomaly detection, because it takes into account all instances, normal or anomalous, that

are correctly classified. Thus, in an unbalanced dataset, such as most anomaly detection cases, it might give a high result even though all anomalies are considered as normal. Considering NAB dataset, if all cycles were considered normal, the accuracy would be approximately 97%, which would be promising. Therefore, the most adequate metrics to assess the quality of anomaly detection algorithms or any problem with unbalanced datasets, are the specificity, sensitivity, precision and, particularly, F1 score, which is the harmonic mean of precision and sensitivity. In this case, F1 score is 90,9% indicating promising results.

4.2.2 Pseudo Periodic Synthetic Time Series

This dataset was constructed with resource to the Anomaly Introduction Framework, so, the anomalies may be distributed across cycles. This makes it harder to label each cycle as normal and anomalous due to various factors. Consider the following situation: the anomaly was introduced in the last half of a cycle and the first half of the next cycle, so what cycle should be considered anomalous? It could be considered both or none, because in practice both cycles contain anomalies, but neither of them *is* anomalous. Now consider the following situation: one cycle has 70% of an anomaly and the next cycle has 30%, so should the second cycle be considered anomalous? It could not be considered anomalous because the majority of the segment is normal, but the anomaly may be so evident that the whole cycle *should* be considered anomalous. These problems have an aggravate, which is the fact that each time series have a low number of cycles, and so, every misclassified cycle would have a great impact on the results.

The solution was to take into consideration each point as a labelled instance. So, although the classification was made in terms of segments, each point was labelled as the label assigned to the segment that contains it. The same process was made in terms of the annotation process, where each point was annotated as normal or anomalous. Note that the outcome of this annotation process is not equal to the annotations provided by NAB because here, every anomalous point is annotated while in NAB only the first point of an anomalous segment is annotated and, therefore, there are no information about the end of the anomalous instance. Therefore, the comparison was made pointwise instead of in terms of cycles, achieving a finer result.

The results are presented in Table 4.4. Similarly to the tuning performed with the NAB dataset, the parameters were chosen in order to optimise the results of the algorithm but, this tuning was made on a small percentage of data, which is more approximate to reality, and those parameters were then used on the whole dataset including the training part. Furthermore, the features were not scaled using the z-normalisation score. The best results were obtained using the details of the wavelet transform, in which the mother wavelet was chosen to be of the family of Daubechies of third order, using $minPts = 5; \varepsilon = 0,001$.

The results are lower than those obtained for NAB dataset. This is due to the fact that

Table 4.4: Results for pseudo periodic synthetic time series dataset.

Metric	Value (%)
Accuracy	$91,6 \pm 5,2$
Specificity	$91,9 \pm 5,1$
Sensitivity	88 ± 26
Precision	52 ± 17
F1 score	64 ± 19
False Negative Rate	12 ± 26
False Positive Rate	$8,1 \pm 5,1$
False Discovery Rate	48 ± 17
Negative Predictive Value	$98,9 \pm 2,5$

the anomalies are not as obvious as the ones in NAB and the fact that the tuning of the parameters was not made taking into account the whole dataset. Furthermore, the low precision may be explained by the fact that the classification was made in terms of cycles, but the anomalies may be spread across different cycles. Thus, if a cycle is not completely anomalous or normal, it will have misclassified points.

4.2.3 MIT BIH arrhythmia database

MIT BIH arrhythmia database is composed of ECG signals with annotations for every heartbeat. These annotations were made by two specialists and regard the position of the R peak of each heartbeat and the correspondent label. Once the developed anomaly detection framework is made to function in a binary way and regarding each time series differently, heartbeats were not classified as shown in Table 4.1. Instead, the most common label of each ECG signal was considered as the normal class, including in signals in which the majority of the heartbeats are arrhythmic.

Furthermore, in order to guarantee a correct segmentation, the R peaks annotations were used to perform it. Therefore, a segment consisted on a portion of a signal from the R peak less 100 data points until the next R peak minus 100 points.

Moreover, it was applied a second order Butterworth band-pass filter in order to attenuate frequencies lower than $1Hz$ and higher than $20Hz$, enabling to reduce interference from normal respiratory frequencies, and muscular and digital noise, respectively.

The results are presented in Table 4.5. The best features were duration, polarity, linear regression and maximum value of each heartbeat. Unlike the first two datasets, and because the number of heartbeats (cycles) per signal is significantly higher than in the previous datasets, it was possible to use the k-NN curve to estimate ε used by the DBSCAN algorithm automatically for each signal, given a fixed $minPts$. Thus, using $minPts = 5$, ε was specific for each signal.

The results show that it is possible to detect anomalous events in ECG signals. The performance metrics values are not as high as in the previous datasets because these are real world electrophysiological signals, which have considerably more complex structures

Table 4.5: Results for anomaly detection in ECG signals from the MIT BIH arrhythmia database.

Metric	Value (%)
Accuracy	89 ± 12
Specificity	$91,6 \pm 9,5$
Sensitivity	82 ± 30
Precision	41 ± 32
F1 score	44 ± 33
False Negative Rate	18 ± 30
False Positive Rate	$8,4 \pm 9,5$
False Discovery Rate	59 ± 32
Negative Predictive Value	$95,9 \pm 9,5$

and in which anomalies may occur in several forms that may be almost imperceptible for untrained eyes. These results are representative about the accuracy score, which is high, but F1 score is low. This means that, the great majority of the dataset is correctly classified, due to the fact that most normal cycles are considered normal, but several anomalous cycles are wrongly classified, which is reflected by the value of sensitivity. Furthermore, false discovery rate is high, indicating that a high number of cycles classified as anomalous are wrongly classified, because they are indeed normal.

4.2.4 Human Motion on industrial scenario

In order to have a full comprehension of the possibilities of application of the developed anomaly detection framework in human motion datasets acquired with resource to inertial sensors, it will be made a detailed analysis of the results achieved with this dataset. Those results will be presented in relation to the cut-off frequency of the low-pass filter in Table 4.6, the comparison between the results achieved with the developed unsupervised segmentation algorithm and the segmentation made with resource to the annotations made during data acquisition in Table 4.7, and to the used features in Table 4.8.

The annotations were made by non specialists, but, similarly to Section 4.2.2, every data point is labelled as normal or anomalous and the analysis is similar to the conducted analysis in that section. Moreover, the beginning and ending of each working cycle is also annotated in order to have results that are not influenced by segmentation algorithms.

Furthermore, once the number of cycles per signal varies widely and there are several signals with a low number of cycles, it was not possible to use the k-NN curve directly in order to estimate them. The followed approach was inspired by the Leave-One-Out validation that is used to test algorithms in the presence of a low number of instances. In this case the approach was to, given N signals, calculate the parameters for every signal, except the signal under evaluation, with resource to the k-NN curve and then use the mean value of the estimated values for the untested signal. This process was repeated to

each signal allowing for an objective test without influence from a human observer.

Table 4.6: Study about the influence of the cut-off frequency selected for the low-pass filter for the HMIS dataset. The values are presented in terms of percentage (%).

	Cut-off Frequency			
Metrics	No filter	10 Hz	1 Hz	0,1 Hz
Accuracy	73 ± 19	72 ± 20	73 ± 19	72 ± 19
Specificity	75 ± 22	74 ± 23	75 ± 22	74 ± 22
Sensitivity	51 ± 45	50 ± 45	53 ± 45	51 ± 46
Precision	16 ± 21	16 ± 22	18 ± 23	17 ± 23
F1 score	18 ± 24	18 ± 24	19 ± 25	19 ± 26
False Negative Rate	49 ± 45	50 ± 45	47 ± 45	49 ± 46
False Positive Rate	25 ± 22	26 ± 23	25 ± 22	26 ± 22
False Discovery Rate	84 ± 21	84 ± 22	82 ± 23	83 ± 23
Negative Predictive Value	$94,5 \pm 8,2$	$94,4 \pm 8,1$	$94,9 \pm 7,7$	$94,4 \pm 8,4$

In Table 4.6, the results obtained using the raw signal, and three different low-pass frequencies used to pre-process each signal prior to classification are reported. The high dispersion of the obtained results does not allow to perform a precise analysis of the results. Nevertheless, given the results, the possible conclusion is that the cut-off frequency of the low-pass filter as a pre-processing step does not influence the results of anomaly detection using the proposed framework. These results correspond to the mean and corresponding standard deviation of the results for each feature applied separately, for each signal and using the annotations to segment the signal.

Table 4.7: Influence of unsupervised segmentation on anomaly detection for the HMIS dataset. The values are presented in terms of percentage (%).

	Segmentation	
Metrics	Annotations	Unsupervised Algorithm
Accuracy	73 ± 19	71 ± 15
Specificity	75 ± 22	74 ± 18
Sensitivity	53 ± 45	52 ± 36
Precision	18 ± 23	20 ± 24
F1 score	19 ± 25	20 ± 20
False Negative Rate	47 ± 45	48 ± 36
False Positive Rate	25 ± 22	26 ± 18
False Discovery Rate	82 ± 23	80 ± 24
Negative Predictive Value	$94,9 \pm 7,7$	$93,4 \pm 8,0$

From Table 4.7 it is possible to study the influence of the unsupervised segmentation algorithm on the task of anomaly detection. The results are regarding the mean and standard deviation value considering all signals and features applied separately. The signals were pre-processed with the application of a low-pass filter with a cut-off frequency of 1Hz. It is possible to conclude that the results are generally not affected by the application of the algorithm when compared to the segmentation performed using annotations.

Table 4.8: Influence of feature selection on anomaly detection for the HMIS dataset. The values are presented in terms of percentage (%). (In the first line, the set of features corresponds to the values of mean, standard deviation, minimum, maximum, IQR, number of peaks, median, kurtosis, skewness, duration, linear regression, zero crossing rate).

Features	Accuracy	Specificity	Sensitivity	Precision	F1 score
Set of features	73 ± 19	74 ± 21	74 ± 35	24 ± 30	30 ± 31
ICA	89 ± 13	96 ± 11	9 ± 28	23 ± 24	6 ± 17
DTW	72 ± 18	77 ± 21	53 ± 44	15 ± 13	16 ± 17
TAM	70 ± 19	70 ± 22	72 ± 41	17 ± 24	24 ± 28
Fourier Transform	74 ± 16	80 ± 19	28 ± 34	14 ± 22	13 ± 17
Polarity	70 ± 17	77 ± 21	38 ± 48	$5,3 \pm 9,3$	8 ± 14
Cumulative Summation	75 ± 20	80 ± 24	28 ± 44	8 ± 12	8 ± 15
Wavelet Approximation	80 ± 23	82 ± 28	40 ± 52	20 ± 26	15 ± 27
Wavelet Details	65 ± 18	64 ± 22	67 ± 47	14 ± 17	20 ± 22
PCA	77 ± 21	78 ± 24	53 ± 47	25 ± 24	22 ± 28
Cosine Similarity	75 ± 20	75 ± 25	53 ± 48	20 ± 17	19 ± 23
AD-PAA	74 ± 18	75 ± 20	59 ± 47	23 ± 29	29 ± 33
Histogram	65 ± 20	65 ± 21	74 ± 32	24 ± 30	30 ± 31
Euclidean Distance	72 ± 22	71 ± 27	58 ± 50	21 ± 17	20 ± 24
Subsegment analysis	67 ± 20	66 ± 21	79 ± 32	18 ± 24	25 ± 28
PCC	69 ± 18	76 ± 23	42 ± 47	9 ± 15	$6,6 \pm 7,3$
PAPR	74 ± 18	76 ± 20	64 ± 40	25 ± 33	29 ± 33

In Table 4.8 the results obtained using each feature described in Section 3.3 are presented. Feature selection is obviously critical for the achieved results, and subsegment analysis may be considered the best feature considering the values of F1 score and sensitivity, though the high variability makes it impossible to give a precise analysis of those results.

Note that the achieved results are low because of one major difference between the process to obtain results with this dataset compared to the others. While in the first two the hyper-parameters used for the DBSCAN algorithm were user-specified in order to optimise the results, and in the MIT BIH arrhythmia database they were directly estimated with resource to the k-NN curve, here, the parameters were selected by a Leave-One-Out approach due to the low number of time series that compose the dataset, and to a reduced number of cycles per signal. Moreover, once data was originated from three different workstations, the mean value of the estimated parameters from different workstations may not represent the correct value for neither of them. Furthermore, each work cycle was labelled by non-specialists and only the more evident anomalies were identified. Thus, the high false positive rate and low precision values may correspond to anomalies that exist, but were not identified in the process of data acquisition and labelling.

Nevertheless, though the error introduced from the process to validate the developed

framework, we may conclude that it is possible to detect anomalies in the analysis of human motion with resource to inertial sensors, because the values of sensitivity, specificity and accuracy are well balanced. Even so, the short amount of data does not allow a full analysis of the application of the proposed solution, because it does not allow to correctly estimate the parameters required for the DBSCAN algorithm.

4.3 Summary

In this chapter it was given a detailed description of each dataset that was used to evaluate the performance of the proposed framework for human motion evaluation and anomaly detection. The results were then presented for each dataset and a detailed analysis was given relative to the aspects thought relevant for anomaly detection regarding human motion monitored using inertial sensors, including the unsupervised segmentation algorithm.

The proposed framework was tested in two datasets composed of artificially periodic constructed signals and two datasets composed of real-world repetitive signals. The achieved results for artificial signals were higher than the obtained for real-world signals, as expected, and it showed to be a promising approach, though its low precision.

Chapter 5

Conclusions

This chapter summarises the developed work as well as the results and achievements of this dissertation. The application in various scenarios is also analysed, and it is proposed the application of the developed work in a specific domain. Furthermore, there will be proposed future steps and guidelines for future research.

5.1 Main Conclusions

Musculoskeletal disorders are a major concern in manufacturing environments due to wrongly executed movements and awkward positions. Most of the tasks executed in those environments are repetitive and using IMUs to measure them, it is possible to acquire repetitive time series with all the information regarding the movements that integrate each task. With this work, it was proposed a new anomaly detection framework in order to detect anomalies on data originated from human movements, specifically, in manufacture environments.

In Chapter 4 it was given a description of the four datasets used to test the possibility of anomaly detection in an unsupervised way in repetitive time series. The results for those datasets were also reported. From those results it is possible to conclude that anomaly detection in general repetitive time series in an unsupervised fashion is feasible, however, at cost of a reduced performance when compared to domain-specific approaches reviewed in the literature. Notwithstanding, a general approach has the value of being easily adapted in order to be applied in different domains and in repetitive time series with different morphologies, such as the case of different workstations in manufacture environments.

In human motion industrial scenarios, which are dominated by repetitive movements,

it was possible to detect anomalies in multivariate time series using accelerometer, gyroscope and magnetometer data. However, the detection depends on the correct feature selection in order to be accurate and still it may present low precision. Moreover, the results from different cut-off frequencies showed that it is irrelevant for the task of anomaly detection.

The proposed unsupervised segmentation algorithm was applied both in NAB and in pseudo periodic synthetic time series dataset with good results regarding anomaly detection, and it was shown that it did not introduce significant error on the HMIS data set, which indicates that it is applicable to cyclic time series. Nevertheless, it is only applicable in cases in which time series are not dominated by anomalies, which may influence the outcome of the segmentation.

From Chapter 4 it is possible to conclude that the most important step for anomaly detection is feature extraction and selection, as it is the step that mostly influences the outcome of the work. Specifically in the developed framework, the appropriate choosing of the hyper-parameters is also critical, as it is clear from the results obtained for HMIS. For the first two datasets those parameters were chosen in order to optimise the results with little concern of practical application and the values of F1 score were 90,9% and approximately 60% for NAB and pseudo periodic dataset, respectively. For the MIT BIH arrhythmia database, the estimation of the hyper-parameters was made automatically and the results decreased, where F1 score was of approximately 40%. For HMIS, once the tests were conducted based on a Leave-One-Out approach, the results for each signal are influenced by the estimation of hyper-parameters from various workstations, which have different morphologies for normal cycles, which may influence the outcome of the results. However, this was the best approach because it is an unsupervised application and the number of cycles per signal is not enough to automatically estimate the hyper-parameter and the optimisation based on achieved results is not applicable in real-life.

While acquiring HMIS data set, it was not possible to acquire video data, which might have helped in post analysis procedures. Once the annotations were made by unqualified spectators based on a visual analysis in *ad hoc* as the workers were performing the designated tasks, the annotations may have been inaccurate regarding work study in its essence, though the variations from the most common movements did in fact occur. Furthermore, due to the long periods of acquisition, the annotations may have been influenced by fatigue.

The developed system could be integrated in a visualisation dashboard, such as the one depicted in Figure 5.1, in order to facilitate post-analysis of the continuous monitoring of each employee.

The monitoring of workers (and people in general) in order to detect anomalies in their behavior, or to produce information of any kind, may be an Orwellian view, but it has already been applied in various contexts, such as marketing or crowd-sourcing solutions, like Google maps and other mobile applications. On a medical scope, the

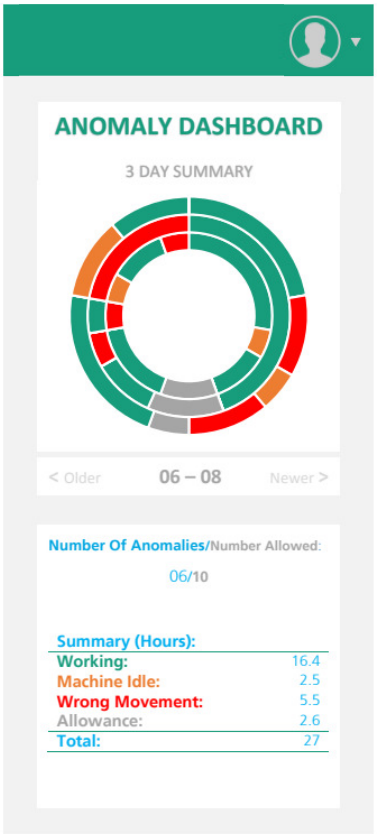

Figure 5.1: Design of an anomaly detection dashboard.

gathered information could be used to benefit the monitored patient and there are various devices that are already used, such as, ambulatory ECG monitoring, smartbands to monitor heart rate and the acitivity of its users, and various other mobile apps with diverse medical applications [52]. Furthermore, the intended application in manufacture scenarios has the goal of prevention of musculoskeletal disorders that may be caused by wrongly performed movements.

5.2 Future Work

The results demonstrated that there is a large margin of progress for the development of a framework for anomaly detection in generic repetitive time series. Further research on anomaly detection should focus on the following topics:

- **Data acquisition** - Though the application of the proposed framework in several data sets, the HMIS data set is not yet large enough in order to have precise results, thus the large values of standard deviation presented in the results. Further data acquisition would also allow to better estimate the hyper-parameters used in DBSCAN algorithm.

- **Feature extraction and selection** - A wide range of features were extracted from each

segmented cycle in order to represent it for the cluster algorithm. However, it is possible to extract more features that may be more representative than the presented in this dissertation. Moreover, a new mechanism capable of correctly selecting the most relevant features prior to the clustering step would be a good asset for the proposed framework. PCA performs this in an unsupervised way, but it does not take into account the inter-variability between different features.

- **Deep learning techniques** - Deep learning techniques are regarded as promising to apprehend the normal behaviour of a data set and forecast future data in case of time series. In literature there are various reports regarding its application on anomaly detection scenarios that show promising results. The main problem, which is the amount of data needed to train deep learning models, may be overcome by data acquisition, and, thus, its application may be possible for anomaly detection on repetitive generic time series, and specifically in the analysis of human movement.

- **Integration of contextual information** - Context is extremely relevant for the task of anomaly detection. The context used in this dissertation is the signal itself. However, in an ambulatory ECG signal, if a patient changed from a seated position to running and then stopped, the variation of heart variability would probability indicate anomalous variations of hear rhythm without context. Nevertheless, if the algorithm took advantage of the context information of the performed tasks, the sudden peak of heart rate would be explainable and might be considered normal, given that the patient was running at the time. This could be applied in various scenarios and could improve the performance of the proposed framework.

- **Development of a visualisation mechanism to present the results** - The results of anomaly detection in manufacturing environments could be used to improve the performance and ergonomy of the monitored employees and, so, an online visualisation mechanism that would provide information in real time could be helpful in that scope. Moreover, it should be investigated the impact of a system of anomaly detection in manufacture environments regarding the appearance of musculoskeletal diseases, which would be a long-term study.

Bibliography

[1] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly Detection: A Survey." In: *ACM Comput. Surv.* 41.3 (July 2009), 15:1–15:58. ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: http://doi.acm.org/10.1145/1541880.1541882.

[2] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity. "Adaptive hidden Markov model with anomaly states for price manipulation detection." In: *IEEE Transactions on Neural Networks and Learning Systems* 26.2 (2015), pp. 318–330. ISSN: 21622388. DOI: 10.1109/TNNLS.2014.2315042.

[3] H. Perista, A. Cardoso, P. Carrilho, and J. Nunes. *Inquérito às Condições de Trabalho em Portugal Continental: Trabalhadores/as*. Tech. rep. Lisboa: Autoridade para as condições do trabalho, 2016.

[4] H. Ren, M. Liu, Z. Li, and W. Pedrycz. "A Piecewise Aggregate Pattern Representation Approach for Anomaly Detection in Time Series." In: *Knowledge-Based Systems* 135 (2017), pp. 29–39. ISSN: 09507051. DOI: 10.1016/j.knosys.2017.07.021. URL: http://linkinghub.elsevier.com/retrieve/pii/S0950705117303465.

[5] E. Keogh, S. Chu, D. Hart, and M. Pazzani. "Segmenting Time Series: A Survey and Novel Approach." In: 2004. ISBN: 9812382909. DOI: 10.1142/9789812565402_0001. arXiv: arXiv:1011.1669v3.

[6] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms." In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. DMKD '03. San Diego, California: ACM, 2003, pp. 2–11. DOI: 10.1145/882082.882086. URL: http://doi.acm.org/10.1145/882082.882086.

[7] L. Wei, N. Kumar, V. Lolla, E. Keogh, S. Lonardi, and C. Ann. "Assumption-free anomaly detection in time series." In: *Siam* (2005), pp. 1–4. URL: http://www.cs.ucr.edu/~wli/publications/WeiL_AnomalyDetection.doc%5Cnwww.cs.ucr.edu/~ratana/SSDBM05.pdf.

[8] D. Haber, A. A. C. Thomik, and A. A. Faisal. "Unsupervised Time Series Segmentation for High-Dimensional Body Sensor Network Data Streams." In: *2014 11th International Conference on Wearable and Implantable Body Sensor Networks*. 1. Zurich, Switzerland: IEEE, 2014, pp. 121–126. ISBN: 978-1-4799-4959-5. DOI: 10.1109/BSN.2014.34. URL: http://ieeexplore.ieee.org/document/6855628/.

[9] M. Weber, G. Bleser, M. Liwicki, and D. Stricker. "Unsupervised motion pattern learning for motion segmentation." In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 202–205.

[10] H. T. T. Thuy, D. T. Anh, and V. T. N. Chau. "Comparing three time series segmentation methods via novel evaluation criteria." In: *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017*. Vol. 2018-Janua. 2018, pp. 171–176. ISBN: 9781538606582. DOI: 10.1109/ICITISEE.2017.8285489.

[11] E. Fink and H. S. Gandhi. "Important extrema of time series." In: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. 2007. ISBN: 1424409918. DOI: 10.1109/ICSMC.2007.4414161.

[12] T.-c. Fu, H.-p. Chan, F.-l. Chung, and C.-m. Ng. "Time Series Subsequence Searching in Specialized Binary Tree." In: *Fuzzy Systems and Knowledge Discovery* (2006), pp. 568 –577. ISSN: 16113349. URL: http://www.springerlink.com/index/t682g4n70360462n.pdf.

[13] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick. "Online segmentation of time series based on polynomial least-squares approximations." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010). ISSN: 01628828. DOI: 10.1109/TPAMI.2010.44.

[14] A. Gensler and B. Sick. "Novel Criteria to Measure Performance of Time Series Segmentation Techniques." In: *Proceedings of the LWA 2014Workshops: KDML, IR, FGWM*. 2014. URL: http://ceur-ws.org.

[15] M. Teng. "Anomaly detection on time series." In: *2010 IEEE International Conference on Progress in Informatics and Computing*. Vol. 1. 2010, pp. 603–608. DOI: 10.1109/PIC.2010.5687485.

[16] Z. Chen and Y. F. Li. "Anomaly Detection Based on Enhanced DBScan Algorithm." In: *Procedia Engineering* 15 (2011). CEIS 2011, pp. 178 –182. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2011.08.036. URL: http://www.sciencedirect.com/science/article/pii/S1877705811015372.

[17] G. Cola, M. Avvenuti, A. Vecchio, G. Yang, and B. Lo. "An On-Node Processing Approach for Anomaly Detection in Gait." In: *IEEE Sensors Journal* 15.11 (2015), pp. 6640–6649. ISSN: 1530-437X. DOI: 10.1109/JSEN.2015.2464774.

[18] E. J.d. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti. "ECG-based heartbeat classification for arrhythmia detection: A survey." In: *Computer Methods and Programs in Biomedicine* (2016). ISSN: 18727565. DOI: 10.1016/j.cmpb.2015.12.008.

[19] A. Vishwa, M. K. Lal, and S. Dixit. "Clasification Of Arrhythmic ECG Data Using Machine Learning Techniques." In: (2011), pp. 68–71. DOI: 10.9781/ijimai.2011.1411.

[20] M. Längkvist, L. Karlsson, and A. Loutfi. "A review of unsupervised feature learning and deep learning for time-series modeling." In: *Pattern Recognition Letters* 42.1 (2014), pp. 11–24. ISSN: 01678655. DOI: 10.1016/j.patrec.2014.01.008. arXiv: 1602.07261.

[21] P. Malhotra, L. Vig, P. Agarwal, and G. Shroff. "TimeNet: Pre-trained deep recurrent neural network for time series classification." In: (2017). arXiv: arXiv:1706.08838v1.

[22] A. Singh. "Anomaly Detection for Temporal Data using Long Short-Term Memory (LSTM)." Master's thesis. KTH Information and Communication Technology, 2017.

[23] B. Sookdeo. "AN EFFICIENCY REPORTING SYSTEM FOR ORGANISATIONAL SUSTAINABILITY BASED ON WORK STUDY TECHNIQUES." In: *The South African Journal of Industrial Engineering* 27.4 (2016), pp. 227–236. ISSN: 2224-7890. DOI: 10.7166/27-4-1552. URL: http://sajie.journals.ac.za/pub/article/view/1552.

[24] T. Best. "Work Measurement in Skilled Labor Environment." In: *Institute of Industrial Engineers* (2012), pp. 20–26. URL: https://www.iienet2.org/uploadedfiles/SHS_Community/Resources/WorkMeasurementinSkilledLaborEnvironments.pdf.

[25] M. Velho. *Heptasense. Tecnologia portuguesa chega às fábricas da Mercedes*. 2018. URL: https://www.dinheirovivo.pt/fazedores/heptasense-tecnologia-portuguesa-chega-as-fabricas-da-mercedez/ (visited on 03/14/2018).

[26] *Job Management Software and Tracking System*. URL: https://www.totalcontrolpro.com/ (visited on 09/20/2018).

[27] K. Bauters, J. Cottyn, D. Claeys, M. Slembrouck, P. Veelaert, and H. van Landeghem. "Automated work cycle classification and performance measurement for manual work stations." In: *Robotics and Computer-Integrated Manufacturing* 51.December 2017 (2018), pp. 139–157. ISSN: 07365845. DOI: 10.1016/j.rcim.2017.12.001. URL: http://linkinghub.elsevier.com/retrieve/pii/S0736584516303787.

[28] J. Evan, Cohn. "Ultrasonic Bracelet and receiver for Detecting Position in 2D Plane." Pat. US 9881276 B2. 2018.

[29] N. Vignais, F. Bernard, G. Touvenot, and J.-C. Sagot. "Physical risk factors identification based on body sensor network combined to videotaping." In: *Applied Ergonomics* 65 (2017), pp. 410 –417. ISSN: 0003-6870. DOI: https://doi.org/10.1016/j.apergo.2017.05.003. URL: http://www.sciencedirect.com/science/article/pii/S0003687017301060.

[30] D. Folgado. "Measuring Repetitive Tasks using Inertial Sensors." Master's thesis. Faculdade de Ciências e Tecnologias - Universidade Nova de Lisboa, 2015.

[31] F. Wang, B. Dun, X. Liu, Y. Xue, H. Li, and Q. Han. "An Enhancement Deep Feature Extraction Method for Bearing Fault Diagnosis Based on Kernel Function and Autoencoder." In: *Shock and Vibration* (2018), pp. 1 –12. ISSN: 10709622. URL: http://widgets.ebscohost.com/prod/customerspecific/ns000290/authentication/index.php?url=https%3a%2f%2fsearch.ebscohost.com%2flogin.aspx%3fdirect%3dtrue%26AuthType%3dip%2ccookie%2cshib%2cuid%26db%3da9h%26AN%3d128215630%26lang%3dpt-br%26site%3deds-live%26scope%3dsite.

[32] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic. "Big-data-driven anomaly detection in industry (4.0): An approach and a case study." In: *2016 IEEE International Conference on Big Data (Big Data)*. 2016, pp. 1647–1652. DOI: 10.1109/BigData.2016.7840777.

[33] G. Singh. *7 methods to perform Time Series forecasting*. 2018. URL: https://www.analyticsvidhya.com/blog/2018/02/time-series-forecasting-methods/ (visited on 09/20/2018).

[34] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. 1991, pp. 1–25. ISBN: 9780262018029. DOI: 10.1007/SpringerReference_35834. arXiv: 0-387-31073-8. URL: http://link.springer.com/chapter/10.1007/978-94-011-3532-0_2.

[35] X. Zhu and A. B. Goldberg. "Introduction to Semi-Supervised Learning." In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3.1 (2009), pp. 1–12. ISSN: 1939-4608. DOI: 10.2200/S00196ED1V01Y200906AIM006. arXiv: 1412.6596.

[36] *Comparing different clustering algorithms on toy datasets*. URL: scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html# (visited on 09/20/2018).

[37] T. Segaran. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. 2007, p. 334. ISBN: 0596529325. DOI: 10.1016/j.jconhyd.2010.08.009. arXiv: arXiv:1011.1669v3. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0596529325.

[38] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016, pp. 151–152. URL: http://www.deeplearningbook.org.

[39] D. Steinley and M. J. Brusco. "A New Variable Weighting and Selection Procedure for K-means Cluster Analysis." In: *Multivariate Behavioral Research* 43.1 (2008), pp. 77–108. ISSN: 0027-3171. DOI: 10.1080/00273170701836695org/10.1080/00273170701836695. URL: http://www.tandfonline.com/action/journalInformation?journalCode=hmbr20.

[40] G. Lee, R. Gommers, F. Wasilewski, K. Wohlfahrt, A. O'Leary, H. Nahrstaedt, and Contributors. *PyWavelets - Wavelet Transforms in Python*. 2006. URL: https://github.com/PyWavelets/pywt (visited on 09/21/2018).

[41] H. Ren, X. Liao, Z. Li, and A. Ai-ahmari. "Anomaly detection using piecewise aggregate approximation in the amplitude domain." In: (2018), pp. 1097–1110. DOI: 10.1007/s10489-017-1017-x.

[42] D. Folgado, M. Barandas, R. Matias, R. Martins, M. Carvalho, and H. Gamboa. "Time Alignment Measurement for Time Series." In: *Pattern Recognition* 81 (2018), pp. 268–279. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2018.04.003. URL: https://doi.org/10.1016/j.patcog.2018.04.003.

[43] I. T. Jolliffe. *Principal Component Analysis, Second Edition*. Second. Vol. 30. 3. Springer, 2002, pp. 1–9. ISBN: 0387954422. DOI: 10.2307/1270093. arXiv: arXiv:1011.1669v3. URL: http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa501/full.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[45] M. Daszykowski and B. Walczak. "Density-Based Clustering Methods." In: *Comprehensive Chemometrics*. 2010. ISBN: 9780444527011. DOI: 10.1016/B978-044452701-1.00067-3. arXiv: 10.1.1.71.1980.

[46] A. Kassambara. *DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning*. URL: http://www.sthda.com/english/wiki/wiki.php?id_contents=7940 (visited on 09/21/2018).

[47] J. Hawkins, S. Ahmad, S. Purdy, and A. Lavin. "Biological and Machine Intelligence (BAMI)." Initial online release 0.4. 2016. URL: https://numenta.com/resources/biological-and-machine-intelligence/.

[48] A. Lavin and S. Ahmad. "Evaluating real-time anomaly detection algorithms - The numenta anomaly benchmark." In: *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*. 2016, pp. 38–44. ISBN: 9781509002870. DOI: 10.1109/ICMLA.2015.141. arXiv: 1510.03336.

[49] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. "PhysioBank, PhysioToolkit, and PhysioNet." In: *Circulation* 101.23 (2000), e215–e220. DOI: 10.1161/01.CIR.101.23.e215. eprint: https://www.ahajournals.org/doi/pdf/10.1161/01.CIR.101.23.e215. URL: https://www.ahajournals.org/doi/abs/10.1161/01.CIR.101.23.e215.

[50] J. Hall. *Guyton and Hall Textbook of Medical Physiology 12th edition*. twelfth. Saunders Elsevier, 2010, pp. 121–123; 143–153. ISBN: 9781416045748. DOI: 10.1007/s13398-014-0173-7.2. eprint: arXiv:1011.1669v3.

[51] E. J.d. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti. "ECG-based heartbeat classification for arrhythmia detection: A survey." In: *Computer Methods and Programs in Biomedicine* 127 (2016), pp. 144–164. ISSN: 18727565. DOI: 10.1016/j.cmpb.2015.12.008. URL: http://dx.doi.org/10.1016/j.cmpb.2015.12.008.

[52] V. L. Anthony Berauk, M. K. Murugiah, Y. C. Soh, Y. Chuan Sheng, T. W. Wong, and L. C. Ming. "Mobile Health Applications for Caring of Older People." In: *Therapeutic Innovation & Regulatory Science* (2017). ISSN: 2168-4790. DOI: 10.1177/2168479017725556.