UNIVERSITÀ DEGLI STUDI DI PADOVA

SCUOLA DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE

TESI DI LAUREA MAGISTRALE

Machine Learning approach to sport activity recognition from inertial data

Relatore: Prof. Angelo Cenedese

Correlatore: Dott. Gian Antonio Susto

Laureando: Luca Minetto
1063982

ANNO ACCADEMICO: 2015-16

ABSTRACT

In this thesis we consider an Activity recognition problem for Cross-Country Skiing; the goal of this work is to recognize different types of Cross Country techniques from inertial sensors equipped on a wearable device.
We want to apply the SAX technique to the acceleration signals, specifically on the Atomic Gestures extracted from them. SAX has been introduced in 2003 and it has been used in several fields of application. Applying SAX we work on time-based features strictly related to the time series, that imply some advantages; precisely, SAX representation allows dimensionality and numerosity reduction and it also allows distance measures to be defined on symbolic approach. We want to find some template which are the signals that best represent the aforementioned main techniques and, using them in the SAX Distance calculation, being able to recognize which activity an athlete is performing.

The thesis is organized as follows:
In Chapter 1 we introduce the argument relating to activity recognition from a scientific and commercial point of view, but in the following we focused on Sports Activity. In Chapter 2 and 3 we describe the SAX technique and the dataset in exam, respectively. In Chapter 4 we provide a recognition algorithm based on SAX that works on Atomic Gestures and in Chapter 5 we test it evaluating the classification accuracy. In Chapter 6 we test the aforementioned algorithm on a well-known dataset of Normal Day Activity. In Chapter 6.5 the conclusion of this thesis.

Alla mia famiglia e
agli amici.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Gesture recognition refers to the mathematical interpretation of human motions using a computing device. Gesture recognition, along with facial recognition, voice recognition, eye tracking and lip movement recognition are components of what developers refer to as a perceptual user interface (PUI). The goal of PUI is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline known as usability.
Initially, in this thesis we extend the gesture recognition's concept to a another more generic: Activity recognition. For this reason, in the following, when we discuss about Activity we use this as a generic term that can be referred to a set of gestures or a single gesture depending on the context.

1.1 OVERVIEW OF ACTIVITY RECOGNITION

Activities recognition is a vast topic therefore a first categorization that simplifies the concepts that will be discussed next, has been introduced in Table 1.

Device	**Sensor**	**Position**	**Area**
Wearable	Acc.	Untouched	Normal Day Activity
Smartphone	Gyr.	Hand-Wirst	Command Gesture
Vision System	Mag.	Arm	Sports/Dance
Dedicated Device	GPS	Waist	Games
	Visual	Foot-Ankle	

Table 1: First simple key-concepts used to categorize the related work.

First of all, it is possible to specify device and sensors used for every single gesture recognition problem. The use of gesture as a natural interface serves as a motivating force for research in modeling, analyzing and recognition of gestures [10]. Indeed vision systems are widely used in most of previous work on gesture recognition. However the performance of such vision-based approaches depends strongly on the lighting condition and camera facing angles, which greatly restricts its applications in the smart environments. Smart environments is still a very important argument because, for example, smart home make it possible to monitoring the well-being of an oc-

cupant in a home [11] [12] [13] and to reduce energy usage [14] [15] [16]. This category includes also Microsoft Kinect that has been a key driver for a new form of hands free interaction. As a low-cost and widely available approach to human motion sensing, the Kinect and the associated open source libraries have enabled researchers to target a range of next generation novel multimedia applications, for example [17]. Kinect is principally equipped with a vision-sensor but the other sensors are used to improve the performance also in difficult conditions such as night vision.
In more recent works reported in literature, the most common devices are smartphones and wearables such as smart-watch or smart-band. In fact, with the rapid development of the MEMS (Micro Electrical Mechanical System) technology, people can easily wear or carry on this kind of sensor-equipped devices in daily life [18]. Typically, these devices are equipped with 3D-accelerometer, gyroscope, magnetometer (well-known as "inertial sensors") and GPS. On gesture recognition, some related works need only accelerometers data [19] [20] [21] [22] [23] [24] [25] [26] [27], in [28] [29] [30] [31] accelerometers, gyroscope and magnetometer data are combined. In particular we can conclude that in all the work just cited the accelerometer is the main sensor while gyroscope and magnetometer are used to improve the gestures recognition. Only few works in literature require also the GPS data [32]. There is a fourth category of devices which is "dedicated devices". This category includes all devices developed for perform only particular tasks. For example Maxxyt[30] which is an autonomous device that can count repetitive movements during strength training in real-time or UbiFinger[33] that realizes operations of real-world devices with gestures of fingers.

The device's position, and then the sensors' position, is another important aspect to take into consideration. A primary distinction has been made between *untouched* and *Body position*. *Untouched* means that the device is located near the subject's body. It isn't attached to the body so it doesn't move with the user. For this reason, the category *untouched* includes the majority of vision systems. In fact, in literature there are many works on home automation that use external camera network to recognize gestures [12] [34]. However, there are few examples where the body gesture input system doesn't depend on external cameras, such as Cyclops[35], a single-piece wearable device that sees its user's whole body postures.
Body position indicates that the sensors are attached to the user's body, in one or more precise locations. This category includes all the wearable-devices. The most significant positions that can be found in literature are: *arm* [31] [36], *hand-wrist* [19] [37] [38] [39] [22] [40] [25] [36], *waist* [41] [35] [26] and *foot-ankle* [28] [42]. In some cases we need to recognize a lot of different gestures so it's necessary to

place different sensors in different positions; in this way the amount of data is increased and the recognition task can be more easier. Any example can be [32] or [29] where continuous user motion is acquired from a body-worn network of inertial sensors. Obviously, the sensors' position is closely related to the Activity and Gesture that you want to recognize. In Figure 1 relationship between *Body Position* and *Area* found in the related work.

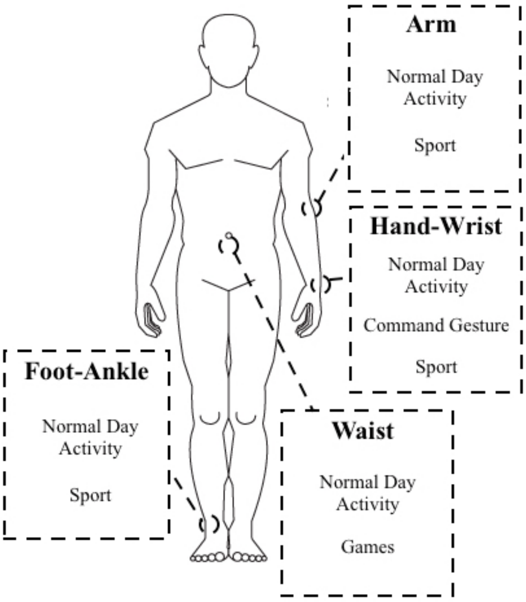


Figure 1: Relationship between *Area* and *body position* found in the related work.

Activity recognition can be used on numerous fields of application then a macro differentiation between different area of application is shown in Table 1.

Normal day activity includes all the activity like walking, sitting, eating [32] [26] [27], cooking, sleeping [11], cycling, driving [41], writing.

Command Gestures, as the name suggest, are all the gestures that have the purpose to command something. This kind of gesture is very common in Home Automation [43]. In this context a specific task can be assigned to a particular gesture and so, for example, a window can be opened by an hand-circular movement.

Sport and dance includes all the works that have the purpose to recognize and assess athlete's activities [44] and in certain case provide feedback on the quality of movements, such as the swing [45] or dance steps [28].

The last category is *Games*. Then, for example, in [17] has been developed a gesture-based game for deaf/mute people using Microsoft Kinect. However Kinect is not the only device used in this context. For example, in [46] has been used a wearable sensors for real-time recognition tasks in games of martial arts [21] or with Cyclops [35], worn at the center of the body, the user can play a racing game on the mobile phone with hand and foot interactions.

Now, we have to introduce a more precise definition of what kind of activity we want to analyze. In Table 2 it can be notice that at different *Activity Type* we have the possibility to face different *problem stage*. *Continuos-Repetitive* means that there is a continuos data stream which admits periods of non-activity amidst periods of activity. In this case the activity is composed by repetitive gestures such as weight training and calisthenics [44] or other sport activities [47]. Instead, in *Continuos-Spot*, the Activity in the continuos data stream is composed by spot gestures. Any example of spot gestures are sitting or standing up [32]. On the other hand, *Isolated* means that the data stream includes only an Activity period and we exactly know when the Activity starts and when it stops. It's simple to conclude that this Activity type includes all command gestures previous cited. Anyway [43] provides an exhaustive example.
In any cases, the three *Possible Problem Stages* in Table 2 are not exactly noticeable; in particular, the first and the second phase can be joined in an unique phase. However, an idea of what every stage represent may be the following:

1. **Finding Activity amidst periods of non-Activity.** This is the first stage for Activity type with a continuos data stream. In literature, it's possible to find some example of segmentation techniques used for solve this problem [48]. For example, in [44] is used a 5-second sliding window, for each window, 224 features is computed, and the resulting feature matrix is used to train an L2 linear support vector machine which predicts either "exercise" or "non-exercise" for each 5-second window. An Automatic segmentation technique based on Signal Energy is provided in [49]. In [50] the "Activity finding" procedure is based on the observation that humans tend to keep their hands in rest positions and any gesture starts from one rest position and terminates at another rest position.

2. **Recognizing the Activity.** This is the main stage for every *Activity Type* and it is necessary for understand which gesture is being performed. To recognize a gesture from the captured data, researchers have applied diverse machine learning, pattern recognition techniques:
   - Dynamic Time Warping (DTW) [51]
   - Hidden Markov Model (HMM) [52]
   - Support Vector Machine (SVM) [1] [53] [54] [55]
   - Decision Tree (DT) and Random Forest (RF) [1] [53]

Activity Type	Possible Problem Stage	Example
Continuos-Repetitive	1) Finding Activity amidst periods of non-Activity. 2) Recognizing the Gesture in the Activity. 3) Counting the repetitions.	We consider a swimmer in a pool. First we want to know when he is swimming and when he is stopped in rest position. Second, for each Activity period, we want to recognize what style he is swimming. Third, we want to know how many strokes he does for each Activity period.
Continuos-Spot	1) Finding Activity amidst periods of non-Activity. 2) Recognizing the Activity.	We consider a tennis player during a match. First we want to know when he is stopped in rest position or when he is servicing, he is smashing,... For each Activity period, we want to recognize exactly what stroke fundamental has been executed.
Isolated	1) Recognizing the Gesture in the Activity.	We consider this situation: an user is subject to a computer questionnaire. The user can answer to each question only with a body gestures. So, after each questions appear on computer's screen, an user's movement will be performed and this gesture must be recognized to proceed with other questions.

Table 2: Possible Problem Stages for different Activity Type.

- Bayesian networks (BN)
- K-Nearest Neighbor (k-NN) [1] [53]
- String Matching (SM) [56] [2] [57]

In Table 3 we present a categorization of related works based on technique employed. In any case, in this stage, after the gesture recognizing it's provided a "gesture's characterization". For example, we recognize a crawl armful in a swimming Activity and then we provide a quality report of the movement such as "overly energetic", "too slow", "too fast", etc.

3. **Counting the ripetitions.** This stage is needed only if an Activity involving repetitive movements. In particular, when an activity is detected and recognized we need to count how many repetitions of this movement there are in this Activity. For example the Counting challenge has been fronted with "peak detection". Precisely, in [30], after a comparison of particular smoothing techniques follow that the smoothed signal's peaks correspond to each repetition in the Activity. Also with RecoFit [44] this stage is well delineated.

Technique	Ref.	Year.	Sensor
DTW	[58]	2015	Kinect
	[47]	2014	Acc. Gyr.
	[34]	2014	Vision
	[22]	2012	Acc.
	[24]	2011	Acc.
	[23]	2009	Acc.
HMM	[59]	2015	Kinect
	[31]	2013	Acc. Gyr.
	[40]	2010	Acc.
	[12]	2009	Vision
	[20]	2006	Acc.
SVM	[44]	2014	Acc.
	[43]	2014	Acc.
	[18]	2009	Acc.
	[21]	2008	Acc.
	[60]	2006	Acc.
DT/RF	[50]	2014	Acc. Gyr.
	[41]	2009	Acc.
	[32]	2008	Acc. Gyr. GPS
BN	[60]	2006	Acc.
	[19]	2004	Acc.
k-NN	[43]	2014	Acc.
	[61]	2012	Vision
	[46]	2006	Acc. Gyr.
SM	[29]	2012	Acc. Gyr.

Table 3: Categorization of related works based on technique employed. Note that Acc. and Gyr. stand for accelerometer and gyroscope, respectively.

1.2 MAIN TECHNIQUES

Before discussing about how each single technique works, it's important to distinguish what *algorithm input* each technique needs to work properly. In Figure 2 it's possible to see that some techniques used for recognizing data work with time-series data or time-based features. In fact, usually, DTW is used directly with the sensor's output signal while SAX works with a particular signal representation that can be intended as feature sequence. Anyway this type of features is strictly connected with the original signal time sequence. For this techniques see Section 1.2.2. Instead other techniques work with signal's features, like signal's energy, that lose the time sequence of the signal. For these reason, techniques like k-NN, SVM, DT/RF firstly need a setup phase commonly named "feature extraction". For this techniques see Section 1.2.1. The term *Other* in Figure 2 includes all the classification techniques that will be not discussed in the following chapters because considered less important in this thesis. For example *Other* includes also Bayesian Network.

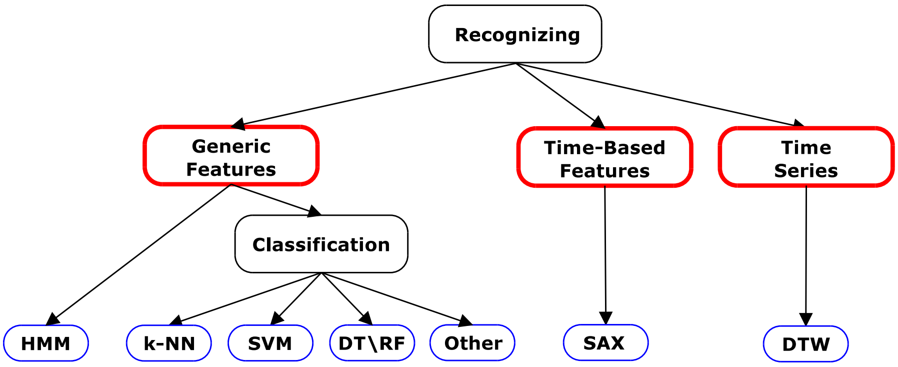


Figure 2: Diagram of employed techniques. *Red edge:* Algorithm input. *Blue edge:* Techniques.

1.2.1 *Classification Techniques*

In gesture recognition context response variables (output variables) take on values in one of K different classes, or categories. Examples of qualitative variables may be swimming style (crawl, butterflystroke, breaststroke, backstroke and medley) or a person activities like walk or run. We tend to refer to problems with a qualitative response as classification problems. In that kind of problem we have a set of training observations $(x_1, y_1), ..., (x_n, y_n)$ that we can use to build a classifier. Where $x_i = (x_{1i}, x_{2i}, ..., x_{pi})$ means that we're considering p features while y_i is the response associated to the observation x_i. Table 4 summarizes the major notation used in this section.

Notation	Meaning
K	number of different classes
n	number of observation
x_i	i-th observation, taking value in X
x_0	query point
p	number of features
X_j	Generic feature
x_{ji}	j-th feature of i-th observation

Table 4: Meaning of the major notation used in this section.

1.2.1.1 *k-NN: k-Nearest-Neighbor Classifiers*

K-NN is a *lazy learning method* so the generalization beyond the training data is delayed until a query is made to the system. k-NN classifiers are memory-based, and require no model to be fit. In this way the classification of a query point x_0 is made to the class of the closest training observations. Precisely, given a query point x_0, we find the k training points closest in distance to x_0, and then classify using majority vote among the k neighbors. The concept behind k-NN is shown, for two different value of k, in the example in Figure 3 Despite its simplicity, k-nearest-neighbors has been successful in a large number of classification problems, including handwritten digits, satellite image scenes and EKG patterns.

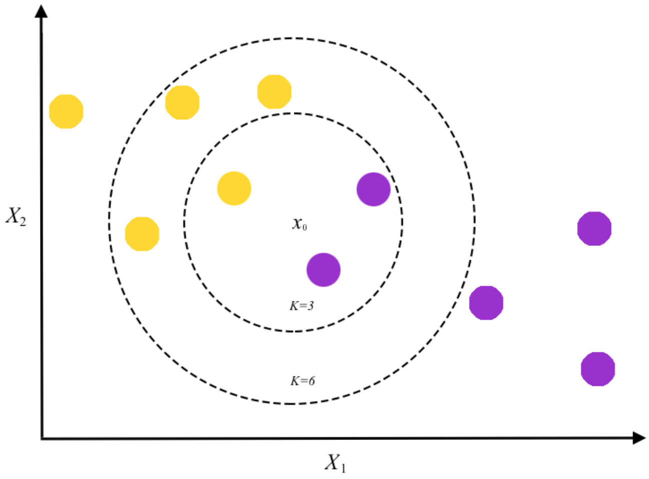


Figure 3: Two classes (blue and in yellow) and $p = 2$. It can be seen the closest point for $k = 3$ and $k = 6$. In the first case, x_0 will be classified as *blue*, in the second case it will be classified as *yellow*.

1.2.1.2 *DT: Decision Tree/RF: Random Forest Classifiers*

Also DT/RF are *lazy learning method*. Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model (like a constant) in each one. In order to make a prediction for a given observation, we typically use the mean or the mode of the training observations in the region to which it belongs. Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are known as decision tree methods.Tree-based methods are simple and useful for interpretation. However, they typically are not competitive with the best supervised learning approaches.

A classification tree is used to predict a qualitative response; we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. It's commonly used recursive binary splitting to grow a classification tree. The *classification error rate* is used as a criterion for making the binary splits. Since we plan to assign an observation in a given region to the most commonly occurring class of training observations in that region, the *classification error* rate is simply the fraction of the training observations in that region that do not belong to the most common class.

Random forests have been developed by Breiman in 2001 and they are a large collection of *decorrelated trees*. Random forests are simple to train and tune. As a consequence, RF are popular and they are implemented in a variety of packages. When used for classification, a Random Forest obtains a class vote from each tree, and then classifies using majority vote.

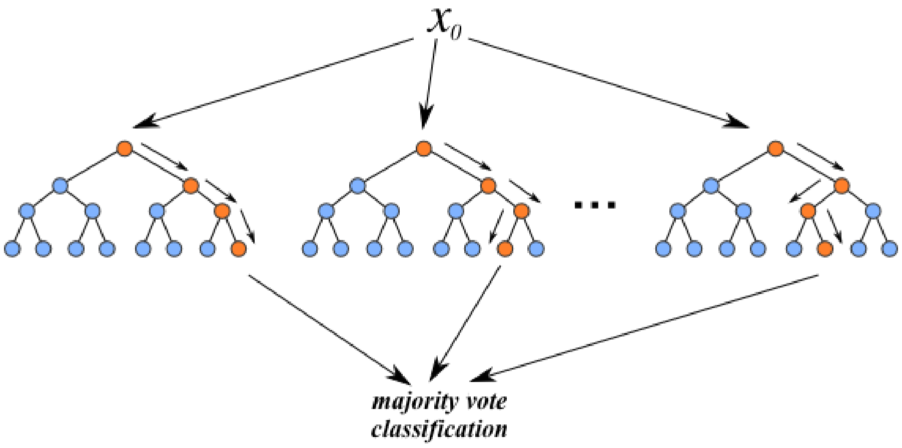


Figure 4: Generic scheme that represent the classification of x_0 using Random Forest. Figure adapted from http://en.likefm.org/artists/images/Random+Forest .

1.2.1.3 *SVM: support vector machine*

Support vector machine is an approach for classification that was developed in the computer science community in the 1990s and that has grown in popularity since then. SVM is a further extension of the *Support Vector Classifier* in order to accommodate non-linear class boundaries. An overview of the Support Vector Classifier is the following.

In Figure 5 is shown an example in which the observations that belong to two classes are not necessarily separable by a hyperplane. In fact, even if a separating hyperplane does exist, then there are instances in which a classifier based on a separating hyperplane might not be desirable because this type of classifier will necessarily perfectly classify all of the training observations; this can lead to sensitivity to individual observations. In fact, the addition of a single observation in the right-hand panel of Figure 6 leads to a dramatic change in the maximal margin hyperplane. The resulting maximal margin hyperplane is not satisfactory for one thing, it has only a tiny margin. This is problematic because the distance of an observation from the hyperplane can be seen as a measure of our confidence that the observation was correctly classified. Moreover, the fact that the maximal margin hyperplane is extremely sensitive to a change in a single observation suggests that it may have overfit the training data. In this case, we might be willing to consider a classifier based on a hyperplane that does not perfectly separate the two classes but guarantees:

- Greater robustness to individual observations
- Better classification of most of the training observations

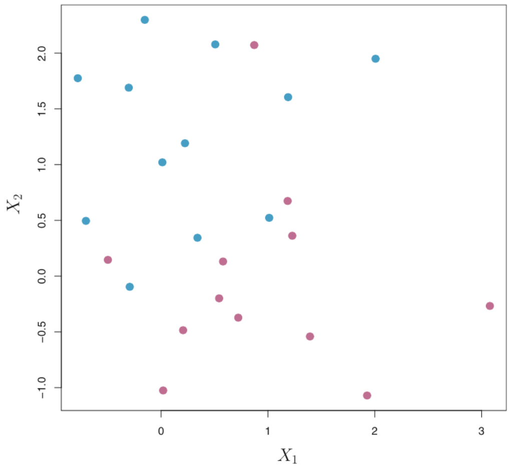


Figure 5: Two classes (blue and purple) and $p = 2$. The two classes are not separable by a hyperplane. Figure adapted from [1].

In this way, the support vector classifier could be worthwhile to misclassify a few training observations in order to do a better job in classifying the remaining observations. Rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane but also on the correct side of the margin, we instead allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane. An observation can be not only on the wrong side of the margin, but also on the wrong side of the hyperplane. In fact, when there is no separating hyperplane, such a situation is inevitable. Observations on the wrong side of the hyperplane correspond to training observations that are misclassified by the support vector classifier.

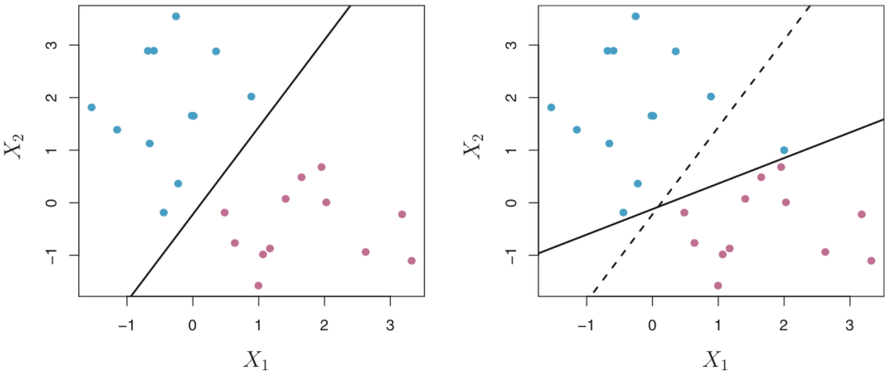


Figure 6: *Left:* Two classes (blue and in purple), along with the maximal margin hyperplane. *Right:* An additional blue observation has been added leading to a shift in the maximal margin hyperplane (solid line). Figure adapted from [1].

The support vector machine classifier is an extension of this idea, where the dimension of the enlarged space is allowed to get very large, infinite in some cases. So far, our discussion has been limited to the case of binary classification: that is, classification in the two-class setting. However we can extend SVMs to the more general case where we have some arbitrary number of classes. The two most popular proposal for extending SVMs to the K-class are the *one-versus-one* and *one-versus-all* approaches.

1.2.2 *Time-dependent Techniques*

Contrary to techniques previously discussed, DTW and SAX have not been developed for classification problem. In the same way aren't born for solve gesture recognition's problem. However they were used often in pattern recognition problems because they don't need feature extraction or other special setup for work properly.

1.2.2.1 *DTW: Dynamic Time Warping*

Dynamic time warping is a well-known technique to find an optimal alignment between two given time-dependent sequences under certain restrictions. Intuitively, the sequences are warped in a non-linear fashion to match each other. Originally, DTW has been used to compare different speech patterns in automatic speech recognition. In fields such as data mining and information retrieval, DTW has been successfully applied to automatically cope with time deformations and different speeds associated with time-dependent data.
The objective of DTW is to compare two time-dependent sequences. These sequences are discrete signals (time-series) or, more generally, continues signals sampled at equidistant points in time. To compare two different signals DTW make use of a local cost measure, sometimes named as local distance measure. Typically, this is "low cost" (small value) if the signals are similar to each other and else is "high cost" (large value). Then DTW search the optimal warping path between the two sequences (signals) which is a warping path having minimal total cost among all possible warping paths. The DTW distance between the two sequences is then defined as the total cost of the optimal warping path. Figure 7 shows a comparison between Euclidean distance and DTW distance.

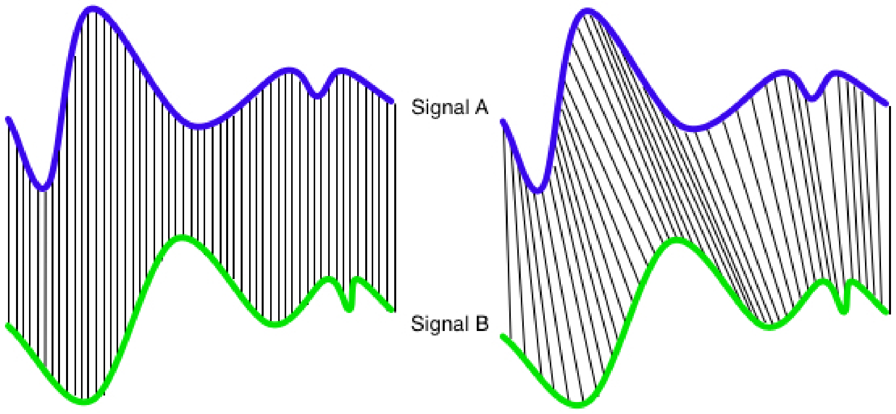


Figure 7: *Left:* Euclidean distance between Signal A and Signal B. *Right:* DTW distance between Signal A and Signal B. Figure adapted from http://www.stanford.edu .

1.2.2.2 *SAX: Symbolic Aggregate approXimation*

In 2003, has been introduced a new symbolic representation of time series. SAX representation allows dimensionality and numerosity reduction, and it also allows distance measures to be defined on the symbolic approach that lower bound corresponding distance measures defined on the original series. SAX allows a time series of arbitrary length n to be reduced to a string of arbitrary length w with $w < n$, typically $w \ll n$. The alphabet size is also an arbitrary integer α, where $\alpha > 2$.The discretization procedure is unique in that it uses an intermediate representation between the raw time series and the symbolic strings. We first transform the data into the *Piecewise Aggregate Approximation (PAA)* representation and then symbolize the PAA representation into a discrete string. There are two important advantages to doing this [2]:

- **Dimensionality Reduction:** We can use the well-defined and well-documented dimensionality reduction power of PAA, and the reduction is automatically carried over to the symbolic representation.

- **Lower Bounding:** Proving that a distance measure between two symbolic strings lower bounds the true distance between the original time series is non-trivial. The key observation that allows to prove lower bounds is to concentrate on proving that the symbolic distance measure bounded from below the PAA distance measure.

The distance between two SAX representations of a time series requires looking up the distances between each pair of symbols, squaring them, summing them, taking the square root and finally multiplying by the square root of the *compression rate* ($\frac{n}{w}$). Figure 8 synthesizes the method to obtain a string from a signal.

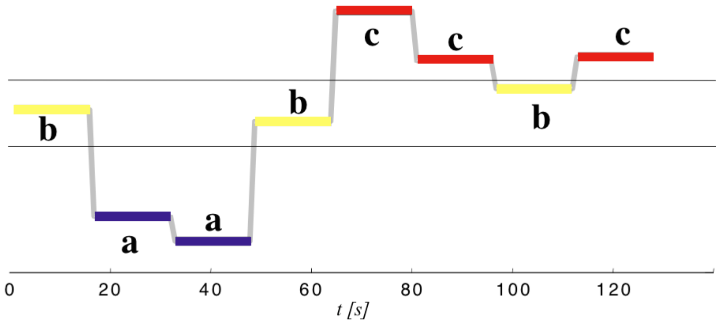


Figure 8: In the example, with $n = 128$, $w = 8$ and $a = 3$. The time series is discretized by obtaining a PAA approximation and then using predetermined breakpoints is mapped map into SAX symbols. In this way, the time series is mapped to the word *baabccbc*. Figure adapted from [2].

1.3 COMMERCIAL PRODUCTS

As already said, recent advancements in microelectronics and other technologies mean that inertial sensors are gaining popularity to monitor human movements in a number of sporting and everyday activities. MEMS sensing technology is already integrated by default into many consumer devices. Virtually every smartphone, smartband and smartwatch are equipped with them. In Table 5 and in Figure 9 and 11, some example of well-known commercial products which use MEMS inertial sensing technology in *Sports and Normal Day Activity Area*.
MEMS inertial sensors are being widely used in motion capture research due to the following reasons [62]:

- They are miniaturized and lightweight so they can be placed on any part or segment of a human body without hindering performance.
- The cost of such sensors is falling dramatically as they start to persuade mass market consumer devices.
- They can be utilized to capture human movement/actions in real unconstrained environments (e.g. outdoor environments with variable lighting conditions) to obtain accurate results.
- They can be used to provide real time or near real time feedback.

Figure 9: *From left to right:* Suunto Ambit3 Peak, Garmin VivoActive, Polar V800, Fitbit Surge. Adapted respectively from [3] [4] [5] [6].

Suunto [3], *Garmin* [4], *Polar* [5] and *Fitbit* [6] are examples of brand of smartwatches and smartbands known in activity tracking, precisely in *Normal Day Activity/Sport Area*. The most of product reported in Table 5 are equipped with few sensors which are not sufficient to perform excellently all the recognition functions in Table 5. For this reason, other sensors is sold separately. For example, *Suunto Ambit3 Peak*®, *Garmin VivoActive*® and *Polar V800*® need a strap with Heart rate monitor to measure the heart rate. For the Activity like Swimming are needed some accessories that permit to recognize the style and measure the cadence of strokes. In the same way, for Cycling *Polar V800*® needs *Cadence Sensor Bluetooth Smart* for pedaling cadence measurement. All this external accessories interact with the main device with Bluetooth technology. More generally, all the device are pedometer-equipped useful for a non-accurate 24-hours recognition of Activity like walking and running with calories consumed reports. In the other hand, *Fitbit*® aims for selling a product already equipped with the sensors useful for recognize the activities reported in Table 5 at the expense of accuracy in some specific function like step cadence. In fact, it can be seen that there aren't sensors labeled as *external*.

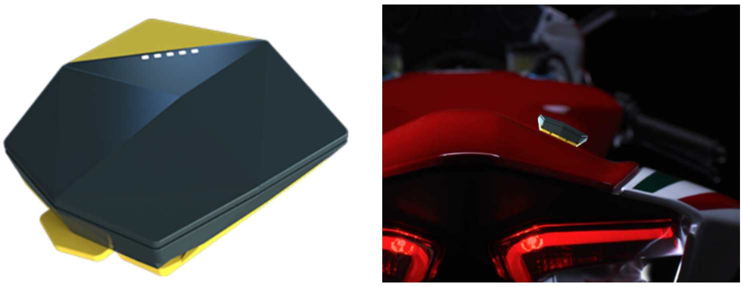


Figure 10: *Left:* Brain one. *Right:* Example of Brain one on a motorcycle. Figure adapted from [7].

Nowadays, *Brain One* [7] is a *dedicated device* for Sport like Gokarting and Motorcycling, but in general it is useful in the majority of *action sports*. It's compact and easy to attach for example on the motorcycle and all its sensors are embedded. Obviously, the heart rate monitor must be attached to the athlete, also the vision system (such as an action camera) is *external* but both interact with *Brain One* with wireless technology. Also *Woo* [9] and *Qlipp* [8] are *dedicated devices* but only for Kite-surfing and Tennis, respectively. Attached to the kite-surf board, *Woo* tracks every jump, including jump height and airtime in a Kite-surf session. At the same way, *Qlipp* attached to the tennis racket, detect and analyze every stroke. All the products aforementioned provide software for post-training characterization useful to improve the training quality.

Figure 11: *Left:* Example of Qlipp attached on a Tennis racket. *Right:* Example of Woo attached on a Kite-surf board. Adapted from [8] and [9].

Device	Inertial Sensors	Other Sensors	Area
SUUNTO AMBIT3 PEAK®	Acc. Gyr. *(external)* Mag.	GPS, Altimeter, Heart rate monitor *(external)*	**Normal Day Activity/Sport:** Cycling, Running, Gym, Free Skiing, Swimming
GARMIN VIVOACTIVE®	Acc. Gyr. *(external)* Mag.	GPS, Heart rate monitor *(external)*	**Normal Day Activity/Sport:** Cycling, Running, Golf, Swimming
POLAR V800®	Acc. Gyr. *(external)* Mag.	GPS, Altimeter Heart rate monitor *(external)*	**Normal Day Activity/Sport:** Running, Fitness exercises, Cycling, Swimming
FITBIT SURGE®	Acc. Gyr. Mag.	GPS, Altimeter Heart rate monitor	**Normal Day Activity/Sport:** Running, Fitness exercises, Cycling
BRAIN ONE®	Acc. Gyr. Mag.	GPS, Barometer Heart rate monitor *(external)* Vision *(external)*	**Sport:** Go Karting, Motorcycling
WOO®	Acc. Gyr. Mag.		**Sport:** Kite-surfing
QLIPP®	Acc. Gyr. Mag.		**Sport:** Tennis

Table 5: Well-known commercial products in *Sports and Normal Day Activity Area*. *External* means that the sensor is sold separately. Note that Acc., Gyr and Mag. stand for accelerometer, gyroscope and magnetometer respectively.

2 SAX: SYMBOLIC AGGREGATE APPROXIMATION

In this chapter we describe in detail SAX, a symbolic representation of time series that has been introduced in 2003.
Since 2003, SAX has been used in several fields of application, such as classification and clustering problems applied on telemedicine time series [63] or entomological problems [64], financial data mining [65] or anomaly detection [66] [67].
Recently, two follow-ups of SAX has been introduced, that are iSAX, used to index massive datasets encountered in science, engineering, and business domains [57] and HOT SAX, used to find unusual time series subsequence [68]. These variants has been introduced principally for indexing and anomaly detection problems.
With this technique we work on time-based features that imply some advantages compared to the other techniques that works with time series, such as DTW. Precisely, SAX representation allows dimensionality and numerosity reduction [2], and it also allows distance measures to be defined on the symbolic approach. Specifically, SAX allows a time series of arbitrary length n to be reduced to a string of arbitrary length w, with $w < n$, typically $w \ll n$; the ratio of n to w is known as *compression rate*. The string of length w is composed by w characters from the alphabet set. The alphabet size is also an arbitrary integer α, where $\alpha > 2$. For example, if $\alpha = 2$ then the alphabet is the set $\{a, b\}$ and an example of string with length 4 can be *abaa*. The discretization procedure is unique: in order to transform the raw time series in the symbolic strings we need an intermediate representation; first, the data is transformed into the *Piecewise Aggregate Approximation (PAA)* representation and then the PAA representation is symbolized into a discrete string; the procedure will be detailed in the following Sections. There are two important advantages to doing this [2]:

- **Dimensionality Reduction:** the dimensionality reduction power of PAA is the well-defined and well-documented [69] [70], and the reduction is automatically carried over to the symbolic representation.

- **Lower Bounding:** Proving that a distance measure between two symbolic strings is lower bound for the true distance between the original time series. The key observation that allows to prove lower bounds is to concentrate on proving that the symbolic distance measure bounded from below the PAA distance measure.

In order to simplify the thesis reading, Table 6 summarizes the major notation used in this section.

Notation	Meaning
n	Time-series length
C	Time-series $C = c_1, ..., c_n$
w	Number of PAA segments representing C
$\bar{C}$	PAA of the time-series C: $\bar{C} = \bar{c}_1, ..., \bar{c}_w$
α	Alphabet cardinality (size)
$\hat{C}$	Symbolic representation of C: $\hat{C} = \hat{c}_1, ..., \hat{c}_w$

Table 6: Meaning of the major notation used in this section.

2.1 PAA DIMENSIONALITY REDUCTION

The concept behind the Piecewise Aggregate Approximation is that a time series C of length n can be represented in a w-dimensional space by a vector $\bar{C} = \bar{c}_1, ..., \bar{c}_w$. In order to do this, the original data is divided into w equal sized *frames* and the mean value of the data, $\bar{c}_i$ falling within the i-th frame is calculated. A visual example of PAA signal's approximation is illustrated in Figure 12.

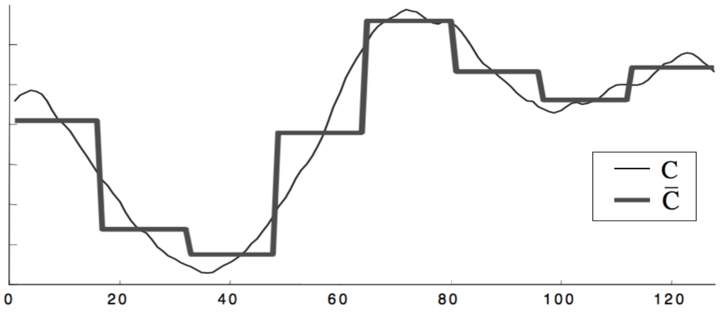


Figure 12: An example of PAA approximation of a signal. C is the original signal while $\bar{C}$ is its *PAA approximation*. Figure adapted from [2].

2.2 DISCRETIZATION

To obtain a discrete representation a further transformation must be applied to the PAA signal. It is desirable to have a discretization technique that will produce symbols with equiprobability. This is easily achieved if we supposed that the time series in exam have a Gaussian distribution; in [2] is demonstrated that the aforementioned assumption is reasonable. For this reason, given an alphabet cardinality α, the *breakpoints* for the discretization can be simply determined by finding

the $\alpha + 1$ points that will produce α equal-sized areas under the Gaussian curve.
Formally, the *breakpoints* are a sorted list of numbers $B = \beta_0, ..., \beta_\alpha$ such that the area under the Gaussian curve from β_i to β_{i+1} is equal to $\frac{1}{\alpha}$. Obviously, β_0 and β_α are defined as $-\infty$ and $+\infty$, respectively.

In brief, a time-series can be discretized in the following manner. First, a PAA is obtained from the original time series and then all PAA coefficients that are below the smallest breakpoint are mapped to the symbol *a*, all coefficients greater than or equal to the smallest breakpoint and less than the second smallest breakpoint are mapped to the symbol *b* etc. The concatenation of these subsequent symbols that represent the signal is called *word*. In Figure 13 an explanatory example of the discretization step is provided.

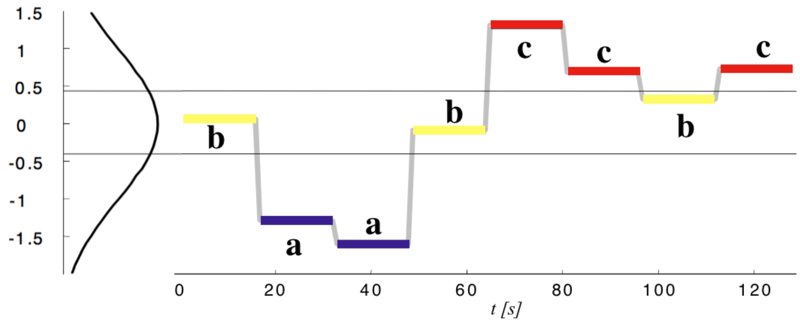


Figure 13: In the example, with $n = 128$, $w = 8$ and $a = 3$, the time series is discretized by obtaining a PAA approximation and then using predetermined breakpoints is mapped map into SAX symbols. In this way, the time series is mapped to the word *baabccbc*. Figure adapted from [2].

The SAX representation procedure can be summarized as follows. Starting from a subsequence $C = c_1, ..., c_n$ let $\bar{C} = \bar{c}_1, ..., \bar{c}_w$ and let α_i denote the i^{th} element of the alphabet. The mapping from the PAA approximation to the correspondent word $\hat{C} = \hat{c}_1, ..., \hat{c}_w$ of length w is obtained as follow:

$$\hat{c}_i = \alpha_j \qquad iif \qquad \beta_{j-1} \leqslant \bar{c}_i < \beta_j \tag{1}$$

2.3 DISTANCE MEASURE

After introducing the *SAX representation*, a new distance measure can be defined on it. The most common distance measure for time series is the *Euclidean distance*, but is not the only one. For the subsequent considerations, let $Q = q_1, ..., q_n$ and C be two time-series of same length n. Let $\bar{C}$ and $\bar{Q}$ represent the *PAA approximation* of C and Q, respectively and $\hat{C}$ and $\hat{Q}$ be their *SAX symbolic representation*. In the following, we detail three distances between the time series in exam.

- **Euclidean Distance:**

$$D(Q, C) = \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2} \tag{2}$$

- **PAA Distance:**

$$D_{PAA}(\bar{Q}, \bar{C}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (\bar{q}_i - \bar{c}_i)^2} \tag{3}$$

- **SAX Distance:**

$$D_{SAX}(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \mathrm{dist}(\hat{q}_i, \hat{c}_i)^2} \tag{4}$$

A visual intuition of these measures is provided in Figure 14 and 15.

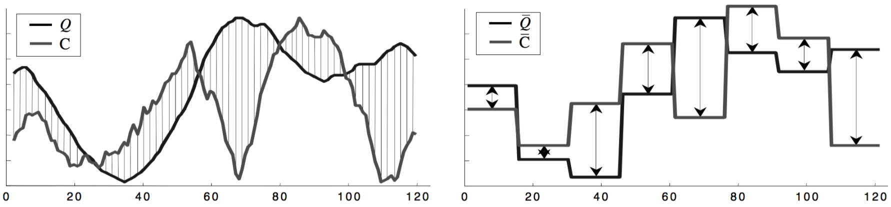


Figure 14: A visual intuition of Euclidean Distance *(left)* and PAA Distance *(right)*. Figure adapted from [2].

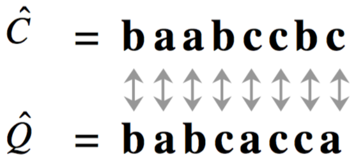


Figure 15: A visual intuition of SAX Distance. Figure adapted from [2].

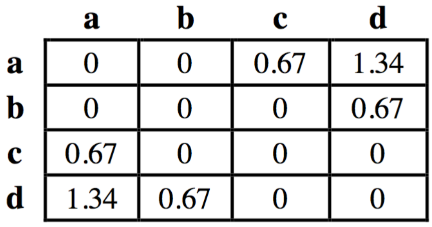
The Equation (3) represent a proved lower bounding approximation of the Euclidean distance between the original subsequences Q and C. The Equation (4) resembles Equation (3) except for the fact that the distance between the two PAA coefficients has been replaced with the sub-function *dist(· , ·)* that weights the distance between couple of symbols. The *dist(· , ·)* function can be implemented using a *lookup table* like the one in Figure 16; it can be noticed that the distance between two adjacent symbols is equal to 0 and the *lookup table* is symmetric. Formally, the value in *cell*$_{(r,c)}$, with r row index and c column index, for any *lookup table* can be calculated by the following expression:

$$\mathrm{cell}_{(r,c)} = \begin{cases} 0 & \text{if } |r - c| \leqslant 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)} & \text{otherwise} \end{cases} \tag{5}$$

The distance between two SAX representations of a time series requires looking up the distances between each pair of symbols, squaring them, summing them, taking the square root and finally multiplying by the square root of the *compression rate*. For example, using the lookup table in Figure 16, the SAX Distance between the two string in Figure 15 is

$$D_{SAX}(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{w}} \; \sqrt{0+0+0+0+0.67^2+0+0+0.67^2}$$

In conclusion it can be noticed that there is a clear tradeoff between the parameter w controlling the number of approximating elements, and the value α controlling the granularity of each approximating element. The SAX technique is highly data dependent thus it's difficult to determine a tradeoff analytically, but it must be found ad hoc empirically.



	a	**b**	**c**	**d**
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

Figure 16: An example of *lookup table* used for determinate the *SAX Distance*. In the example the alphabet cardinality is equal to 4, $\alpha = 4$. The distance between two symbols can be read off by examining the corresponding row and column. Then, for this example, *dist(a,b)=0* and *dist(a,c)=0.67*.

3 CLASSIC CROSS-COUNTRY SKIING DATASET

The activity recognition problem considered in this thesis is composed by 8 athletes that perform 3 particular techniques of *Cross-Country Skiing*. We want to create an activity recognition algorithm that allow to discriminate between these three techniques. In this chapter we provide a brief description of Cross-Country Skiing and its techniques.

3.1 CROSS-COUNTRY SKIING

Skiing started as a technique for traveling cross-country over snow on skis, starting almost five millennia ago in Scandinavia. Cross-country skiing evolved from a utilitarian means of transportation to being a world-wide recreational activity and sport, which branched out into other forms of skiing starting in the mid-1800s. Early skiers used one long pole or spear in addition to the skis.
Cross-country skiing has two basic propulsion techniques, which apply to different surfaces:

- *Classic*, with surface as undisturbed snow and tracked snow.
- *Skate Skiing*, with firm and smooth snow surfaces.

The *classic technique* relies on wax or texture on the ski bottom under the foot for traction on the snow to allow the skier to slide the other ski forward in virgin or tracked snow. With the *skate skiing technique* a skier slides on alternating skis on a firm snow surface at an angle from each other in a manner similar to ice skating. Both techniques employ poles with baskets that allow the arms to participate in the propulsion. Specialized equipment is adapted to each technique and each type of terrain.

In this thesis work it has been considered only the *classic technique of Cross-Country Skiing*. The classic style is often used on prepared trails that have pairs of parallel tracks cut into the snow, but it is also the most usual technique where no tracks have been prepared. Following this technique, each ski is pushed forward from the other stationary ski in a striding and gliding motion, alternating foot to foot. With the *diagonal stride variant* the poles are planted alternately on the opposite side of the forward-striding foot; with the *kick-double-pole variant* the poles are planted simultaneously with every other stride. At times, especially with gentle descents, *double poling* is the sole means of propulsion. On uphill terrain, techniques include the *side step* for steep slopes, moving the skis perpendicular to the fall line, the *herringbone* for moderate slopes, where the skier takes alternating steps with the skis splayed outwards, and, for gentle slopes, the skier uses the diagonal technique with shorter strides and greater arm force on the poles.

In brief the *variant* of classic technique considered in the following are *double poling* (DP), *kick-double-pole* (KDP) and *diagonal stride* (DS) which are illustrated in Figure 17, 18 and 19, respectively.

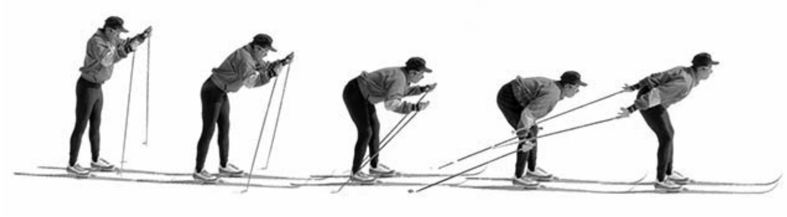


Figure 17: From left to right, the representative steps of Double Poling technique. Figure adapted from *http://skixc.com*.

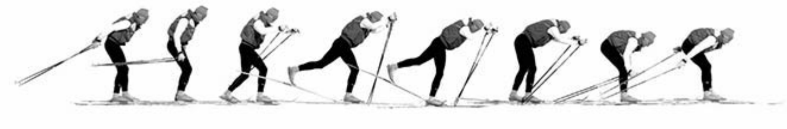


Figure 18: From left to right, the representative steps of Kick-Double-Pole technique. Figure adapted from *http://skixc.com*.

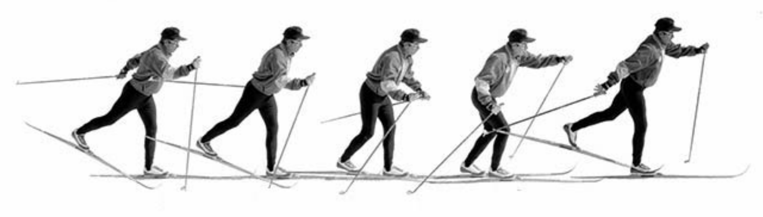


Figure 19: From left to right, the representative steps of Diagonal Stride technique. Figure adapted from *http://skixc.com*.

Given that Cross-Country Skiing is a world-wide recreational activity and sport, this technique can be performed at different *skill levels*. Thus for the 8 athletes considered, in Table 7, is reported the correspondent *skill level* in addition to the athletes' gender. Precisely there are two skill levels: *recreational (R)* and *achiever (A)* and so the dataset consist in four athletes *A* and four *R*. From Table 8 it can be noticed that *Athlete 3* performs only the DP variant and this aspect will be discussed more deeply at the end of this Chapter. All the other athletes perform DP, KDP and also DS. These informations permit to develop some hypotheses on the final results. For example, it is reasonable to think that an achiever athlete performs the different variants of the technique (DP, KDP, DS) better and more consistently than a recreational one.

Athlete	Gender	Skill Level
1	Male	R
2	Female	A
3	Male	A
4	Female	A
5	Female	A
6	Male	R
7	Female	R
8	Male	R

Table 7: Informations about athletes considered in the dataset. Notice that A and R stand for *Achiever* and *Recreational*, respectively.

	Athlete							
Technique	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
DP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KDP	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
DS	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes

Table 8: Cross-Country skiing variants with the athletes that adopt each different technique. Notice that DP, KDP and DS stand for *Double Poling*, *Kick-Double-Pole* and *Diagonal Stride*, respectively.

3.2 CLASSIC CROSS-COUNTRY SKIING ACTIVITY

Referring to the *Activity Categorization* introduced in Section 1.1 in this paragraph we want to describe the Classic Cross-Country Skiing intended as Activity that we want to recognize.
Following the guideline concepts in Table 1, in this work, the *Classic Cross-Country Skiing activity* can be described as reported in Table 9.

Device	**Sensor**	**Position**	**Area**
Wearable	Accelerometer	Hand-Wirst	Sport Activity

Table 9: Categorization of Classic Cross-Country Skiing activity according to the key-concepts introduced in Section 1.1.

Specifically the device used is a smartwatch that permit to collect accelerometer, gyroscope and magnetometer data at sampling frequency 100 Hz while an Athlete is performing Classic Cross-Country Skii. For simplicity, this thesis work we consider only the 3-axial accelerometer data. As Section 1.1 suggests, other activity categorization can be evaluated. Classic Cross-Country Skiing can be classified as *Continuos-Repetitive Activity* given that this Activity turns in continuos data streams which admits periods of non-activity amidst periods of activity. Furthermore the activity is composed by repetitive gestures: *DP gestures*, *KDP gestures* and *DS gestures*.

Now, a definition of *Atomic Gesture* must be introduced:

- **Atomic Gesture:** In this situation we consider Atomic Gesture, one single gesture of type DP, KDP or DS whose phases are respectively illustrated in Figure 17, 18 and 19. So the data stream, apart from non-activity periods, is a succession of Atomic Gesture.

The first aspect analyzed with the Atomic Gesture is the *Period*.

- **Period:** We consider period of the Atomic Gesture, the durate of a single gesture of type DP, KDP or DS; usually it is measured in seconds.

Analyzing the athletes period we can get some conclusions on each athlete and we can notice some difference between DP, KDP and DS.
In Figure 20 we can easily see the mean period of each Athlete for the three different gestures. In this way, we can identify which athlete perform the specific technique faster or slower than the others. For example, *Athlete 6* is the fastest DP performer while *Athlete 7* is the slowest. A general conclusion can be that *Athlete 6* is the fastest in performing all the techniques and the average value calculated in each

Atomic Gesture Class	**Reference Period**	Samples Number (n)
DP	1.50 s	150
KDP	1.60 s	160
DS	1.42 s	142

Table 10: *Reference period* for DP, KDP and DS Atomic Gesture Class measured in seconds and Samples Number, n, that represent the Atomic Gesture length.

class represent the class *reference period* obtained from the average between the mean period of all the athletes. In table 10 is reported the *reference period* for DP, KDP and DS techniques.

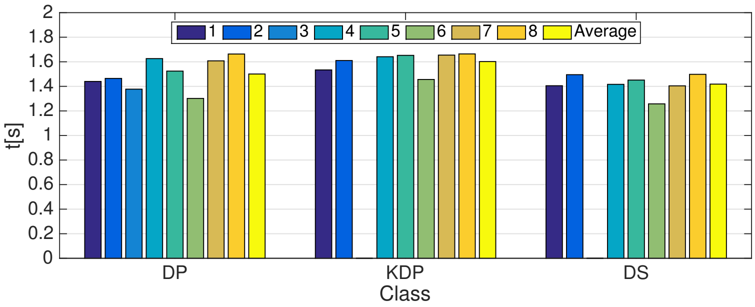


Figure 20: Mean period for all the athletes divided for class. The last bar of each class is the *reference period* for the particular Gesture. Notice that *Athlete 3* haven't data for class KDP and DS.

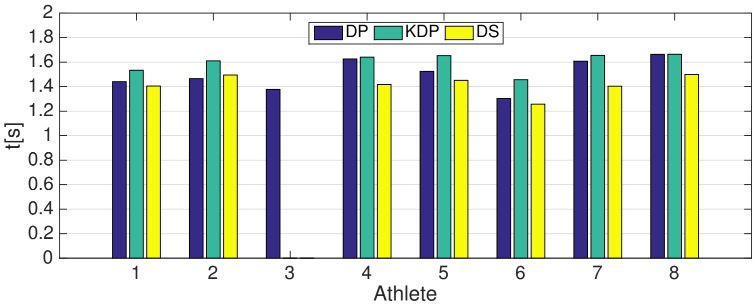


Figure 21: Mean period of each type of gesture divided for athletes. Notice that we don't have KDP and DS data for *Athlete 3*.

The consideration that KDP atomic gesture is slower than the DP and DS is better observable in Figure 21 where is reported the mean period of each type of gesture divided for athletes. It is clear that despite the velocity between the athletes is different, each athlete takes longer to perform a KDP gesture rather than his DP gesture. On the other hand, each athlete takes less time to perform a DS gesture rather than his DP or KDP gesture. However this first analysis don't permit to understand which athlete performs the techniques with a better quality of movement.

In the previous analysis it can be noticed that *Athlete 3* hasn't data associated to KDP and DS techniques. This is a problem for the following consideration. For this person a comparison between gestures can't be provided and thus, also the DP class data, must be considerated inconsistent. For this reason *Athlete 3* data are not included in future discussion. With this dataset restriction the Athletes included in the dataset remain 7, all with consistent data for the three class of Gesture: DP, KDP and DS.

4

SAX TECHNIQUE APPLICATION

4.1 INTRODUCTION

In the previous chapter we have discuss the dataset. The goal of this thesis is to apply the SAX technique, widely described in Chapter 2, on the Classic Cross-Country Skiing dataset and to be able to classify the technique performed by the Athletes. In order to use SAX Distance for classification only one acceleration data has to be used. SAX Distance must be calculated on two strings for which the comparison makes sense. For example, it would not be reasonable to compare a string obtained from x-acceleration data with a string obtained from y or z-acceleration data. Specifically, we consider only the *x-axis acceleration data* because is the most informative signal in the dataset in exam. In fact, the *axes y* and *z* data stream represent a signal with an high noise level. For this reason we chose to not considerate *y* and *z acceleration*. In Figure 22 an example of DP atomic gesture where it is apparent the higher informative content in the x-axis than the other accelerometer signal. In Figure 23 we show a diagram of the proposed algorithm.

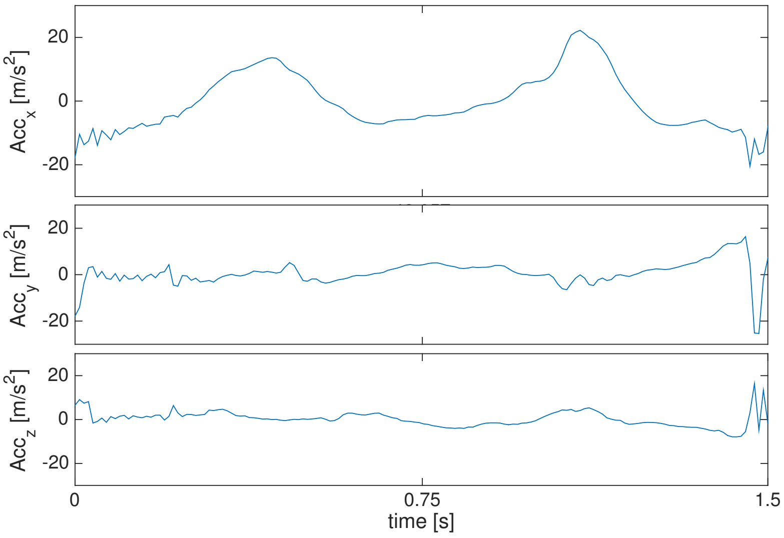


Figure 22: Example of typical DP Atomic Gesture. The highest part of the informative content of the Gesture seems to be contained in the *x-axis*.

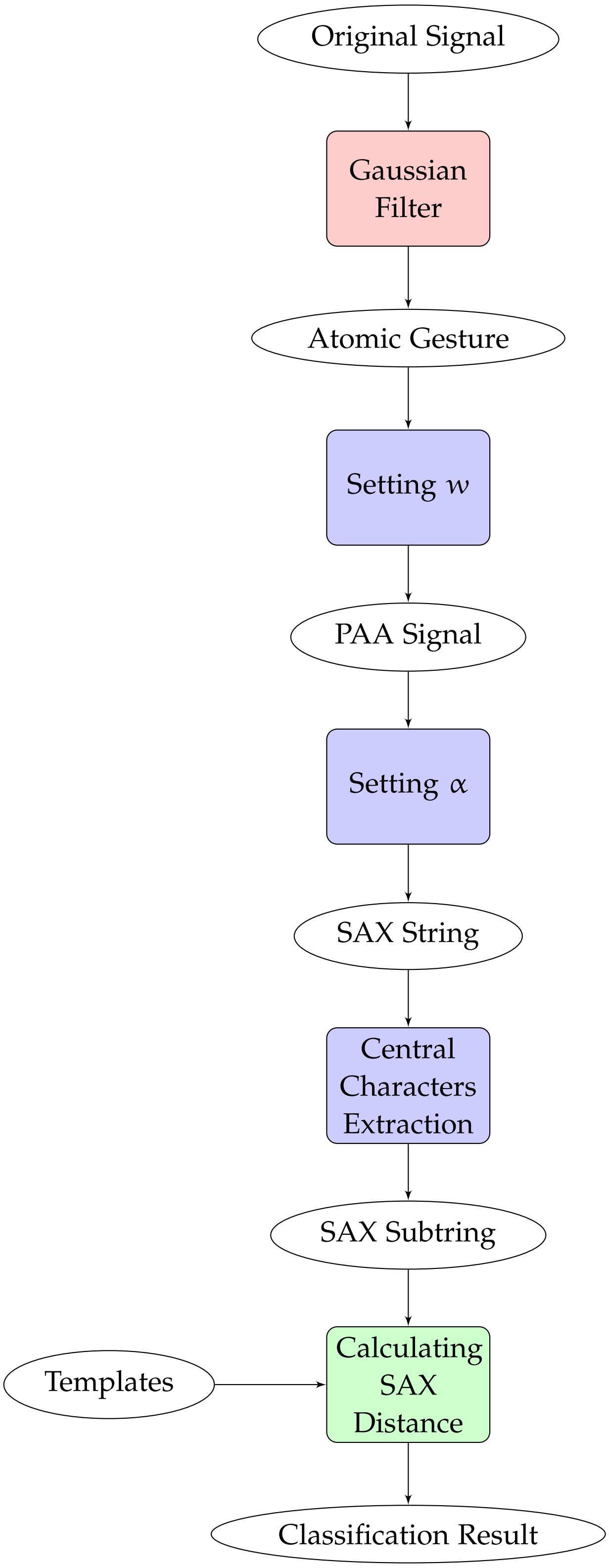


Figure 23: Algorithm diagram. Three color indicate the different phase of the algorithm: in *red*, Gesture Identification (Section 4.2), in *blue*, SAX (Section 4.3, 4.4 and 4.5) and, in *green*, Classification (Chapter 5).

4.2 GESTURE IDENTIFICATION

Starting from the x-acceleration data stream we extract the *Atomic Gestures* of each Athlete using a **Gaussian Filter** created ad hoc [71]. In Figure 24 is illustrated how the Gaussian Filter allows to isolate each single gesture amidst the data stream.

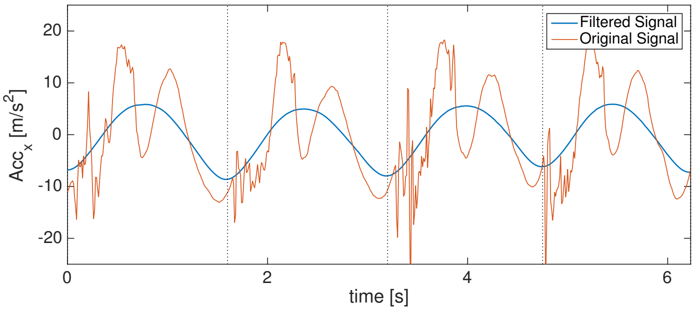


Figure 24: The *original signal* represent the data stream. Filtering the signal with a Gaussian Filter we can individuate the Atomic Gestures. In Figure a *dotted line* separates two consecutive Atomic Gestures.

In brief, we want to apply the SAX technique on all this *Atomic Gestures* and then find the most representative *Atomic Gesture* for each class (DP, KDP and DS). These most representative Atomic Gesture are called *templates* and they will be useful to solve the classification problem with SAX Distance. SAX needs a setting phase in which find optimal values for the parameters w (*word lenght*) and α (*alphabet size*). For this reason Section 4.3 and Section 4.4 details the steps taken to search, respectively, w and α optimal values. While in Chapter 5 are reported some considerations on the templates choice with the correspondent classification results.

4.3 SETTING THE WORD LENGTH

The *Word Length* w is closely connected to the signal PAA. For this reason, we search w in such a way that a trade-off between the two following objectives is achieved:

1. The PAA approximated signal well represents the original signal trend.

2. The PAA approximated signal reduces the noise effect on the signal.

It is clear that increasing w the PAA approximated signal represent more precisely the original signal movement. But we want to maintain w much lower than the Atomic Gesture signal length (n). In Table 11 are summarized some examples obtained varying the value of w;

the signal in the example, depicted in Figure 25, 26 and 27, represent a DP Atomic Gesture so the *reference period* is 1.5 s corresponding to 150 samples (n=150).

Atomic Gesture	**Word Length**	**Compression Rate**	**Figure Ref.**
DP	10	~ 15	Figure 25
	20	~ 8	Figure 26
	30	~ 5	Figure 27

Table 11: DP Atomic Gesture with *reference period* 1.5 s ($n = 150$). Different compression rate are considered. In Figure Ref. is visible the PAA signal for different w values.

The signal shows high noise at the beginning and at the end of the Atomic Gesture. Figure 25, 26 and 27 prove that higher compression rate values imply more noise filtered. On the other hand, the PAA signal results less adherent to the than the original signal. These conclusions can be easily appreciated in Figure 25. In this application we retain more important that the PAA signal evolution follow at the best the original signal evolution thus we chose a length word $w = 30$. A value of w high enough to follow the original signal, but with a reasonable *dimensionality reduction* effect.

In this paragraph we have considered the *reference period* 1.5 s to simplify the argument. Actually, as in Section 3.2, the period is different between different Atomic Gestures also if they are in the same Gesture class and executed by the same Athlete. For example, it's not strange find DP Atomic gesture with period 1.3 s even though the reference period for DP Atomic Gesture is 1.5 s.

4.3.1 *SAX on variable period*

Working on Atomic Gestures it is necessary to consider the *period* information. For this reason, even if the *word length* is fixed, the number of samples considered in each *PAA frame* vary with the signal period. For example, with $w = 30$, if the Atomic Gesture signal length is 150 (period=1.5 s) then the number of samples in each PAA frame is about 5 else if the Atomic Gesture signal length is 120, in each PAA frame there are 4 samples. This is not a problem in the proposed framework because we may think if an Athlete performs a Gesture faster then the signal maintains the same shape but proportionally scaling it.

So the length of each PAA frame is proportional to the Atomic Gesture period, but usually differs from one gesture to another. For this reason we calculate the number of samples in each PAA frame as $\frac{n}{w}$ rounded down to the nearest integer. In most cases n is not

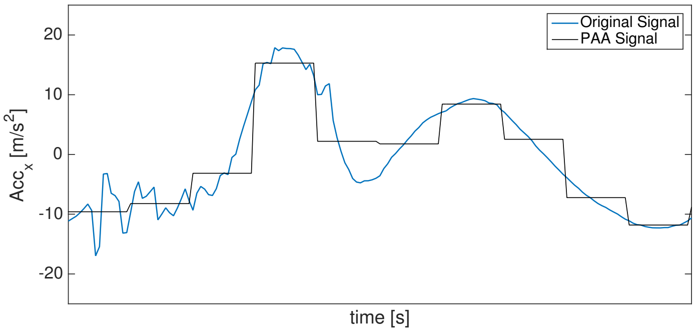


Figure 25: A DP Atomic Gesture. Word length: *w*=10, Atomic gesture period: 1.6 s ($n = 160$).

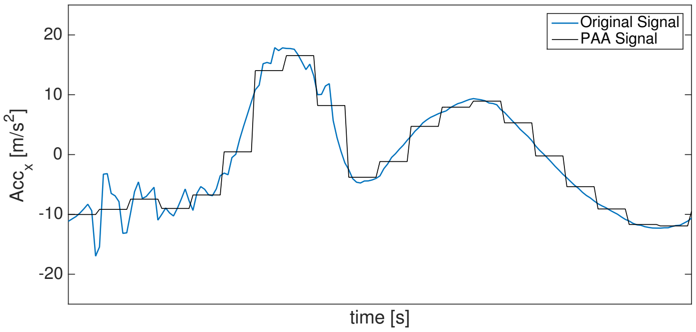


Figure 26: A DP Atomic Gesture. Word length: *w*=20, Atomic gesture period: 1.6 s ($n = 160$).

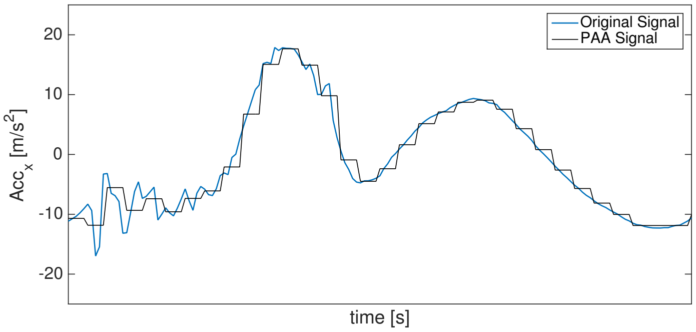


Figure 27: A DP Atomic Gesture. Word length: *w*=30, Atomic gesture period: 1.6 s ($n = 160$). Notice that n is not divisible for *w* then the last *PAA frame* is longer than the other.

divisible by w and then the rest of this fraction indicate the number of samples not considered in the PAA approximation. With $w = 30$ the rest could be an integer in the interval between 0 and 29. At this point, two options have been evaluated :

1. Not to consider the excess samples.
2. To include the samples in excess in the last PAA frames calculation.

With the first solution we have an uncontrollable information loss; in fact we eliminate a signal's portion of length variable from an Atomic Gesture to another. Also the second option has some problems; in this case the last PAA frame includes a variable number of samples, for this reason is different from the other PAA frame in the same Atomic Gesture. By increasing the *word length* the aforementioned problems are accentuated.
To avoid information loss and last PAA frames too different from the others, we use the period information as follow.

1. We calculate the PAA frame length (ℓ) as $\frac{n}{w}$ rounded down to the nearest integer.
2. We calculate the rest of $\frac{n}{w}$, r, that is an integer between 0 and $w-1$.
3. We calculate how many other PAA frames could be created with the r excess samples as $\frac{r}{\ell}$ rounded down to the nearest integer.
4. We calculate the rest of $\frac{r}{\ell}$, that is an integer between 0 and $\ell - 1$.

The last rest represents the real number of samples neglected that, at worst, is still significantly less than $w-1$. In fact, with this method it's obvious that increasing w then the final number of excess samples tends to 0. In this way we get that all of the Atomic Gesture signals could be approximated with a PAA signal formed by $w + \lfloor \frac{r}{\ell} \rfloor$ frames of equal size. In this way, however, the number of frames vary from an Atomic Gesture to another and depends from the period thus leading to having some Atomic Gestures with word length greater than w. This kind of problem will be discuss in Section 4.5 where PAA signal translation into SAX string will be detailed.

4.4 SETTING THE ALPHABET SIZE

The first step to choose the *alphabet size* (α) is fitting the PAA signals obtained as depicted in Section 4.3.1 with a Gaussian distribution. This distribution is strictly connected to the word length and to which Atomic Gestures are considered in the training phase. For example, in Figure 28 we consider separately the DP, KDP and DS Atomic Gestures generating three different Gaussian distribution, one for each Gestures Class. The goal is to have a discretization technique that will produce symbols with equiprobability. The Gaussian distributions just described, however, could well represent only one Gestures Class.

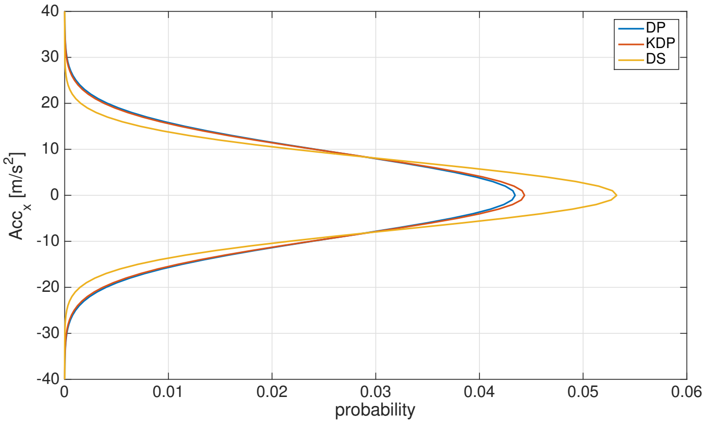


Figure 28: The three different Gaussian distribution, one for each Gesture Class. The *word length* is $w = 30$ in any case.

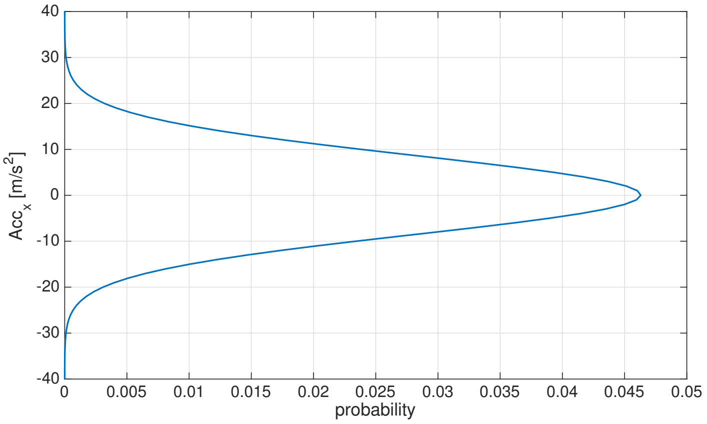


Figure 29: Gaussian distribution of 150 Atomic Gestures (50 for each class). The *word length* is $w = 30$.

It's reasonable to think the three class having equal probability. For this reason, in the following, we consider the Gaussian Distribution generated fitting data from the same number of DP, KDP and DS Atomic Gestures converted PAA approximated. An example of this Gaussian Distribution obtain from 150 Atomic Gestures (50 for each class) is illustrated in Figure 29 and it is characterized by *standard deviation* $\sigma = 8.616$ and *mean* $\mu = 0.067$. Once the reference Gaussian is fixed the *breakpoints* $B = \beta_1, ..., \beta_{\alpha-1}$ can be calculated such that the area under the Gaussian curve from β_i to β_{i+1} is equal to $\frac{1}{\alpha}$. In Table 12 B is reported varying the alphabet size obtained from the Gaussian in Figure 29.

Alphabet size	Breakpoints
6	[-8.26 -3.64 0.067 3.77 8.40]
7	[-9.13 -4.80 -1.48 1.61 4.94 9.26]
8	[-9.84 -5.74 -2.67 0.067 2.81 5.87 9.97]
9	[-10.45 -6.52 -3.64 -1.13 1.27 3.77 6.65 10.58]
10	[-10.97 -7.18 -4.45 -2.11 0.067 2.25 4.58 7.31 11.10]

Table 12: Breakpoints (B) values varying the alphabet size α). This values are obtained from the Gaussian Distribution in Figure 29 with $\sigma = 8.616$ and $\mu = 0.067$.

4.5 DISCRETIZATION PHASE

After establishing the word length and the alphabet size as suggested in the previous Sections, we have to obtain the SAX string from the PAA signal; in order to achieve that, all the PAA coefficients that are below the smallest breakpoint are mapped to the symbol *a*, all coefficients greater than or equal to the smallest breakpoint and less than the second smallest breakpoint are mapped to the symbol *b*, etc. However, as mentioned in Section 4.3.1, the PAA signals have a variable number of coefficients that changes from one Atomic Gesture to another, but is lower bounded by *w*. For this reason after the discretization phase we obtain one *SAX string* for each Atomic Gesture, but the length of this *word* is greater or equal than the *word length* desired. In Figure 30 an example of two DP Atomic Gestures with different period are reported; it can be noticed that, although *w* is the same, the number of PAA frames (coefficients) differs between them, leading to have SAX strings with different lengths. To calculate SAX distance between two strings we need words of the same length; for this reason, we need to extract a particular substring from the SAX string just obtained.

In Section 4.3 we explained that the original signals show a lot of noise at the beginning and the end of the signal itself. A PAA signal obtained with an high value of *w* is preferable but it suffers more this noise problem because it follow better the original signal trend. For this reason the first and the last PAA frames and the correspondent character in the SAX word carry less information than the others. To find the most important substring we proceed in the following way:

1. We first consider an Atomic Gesture with period n that is divisible by w and so we obtain the correspondent word with length w. In this situation we can evaluate how many characters to consider important and we extract them from the SAX string. Given the previous consideration, the most important characters are the *central ones*. These w_1 extracted characters form the substring that represent the Atomic Gesture.

2. At point 1 we found a substring of length w_1. Even if for some Atomic Gesture n is not divisible by w, and so the word is longer than w, we can simply extract the w_1 *central character*.

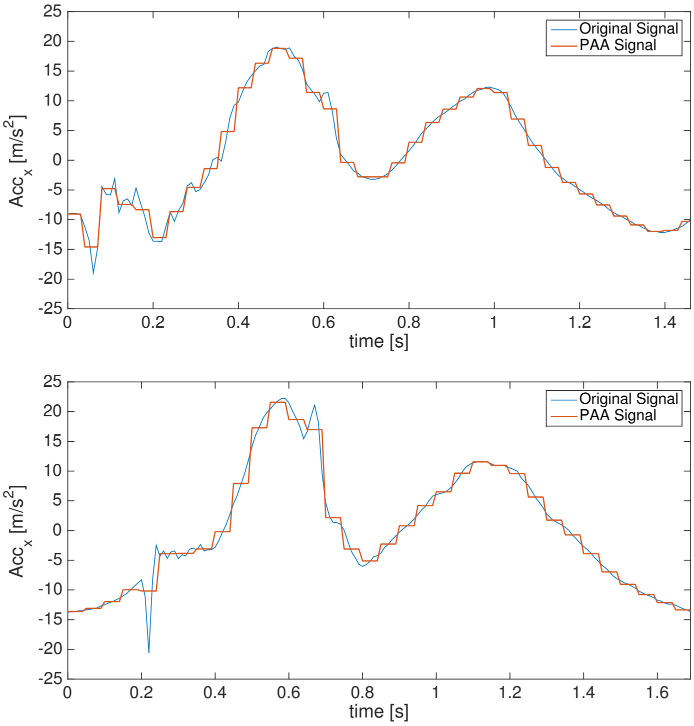


Figure 30: In both examples $w = 30$. *Top panel*, a DP Atomic Gesture with $n = 148$ that is approximated with 37 PAA frames. *Bottom panel*, a DP Atomic Gesture with $n = 170$ that is approximated with 34 PAA frames.

In Figure 31 we provide a visual intuition of the aforementioned procedure with an example of DP Atomic Gesture in which n is not divisible for w. With this procedure we obtain a SAX string that represent better the original signal because w is high and we eliminated the characters correspondent to the portion of original signal more affected by noise and, therefore, less informative. At this point, the SAX distance can be calculated between two substring that have the same length w_1. Being $w_1 < w$ it is required less memory occupation if we need to memorize the SAX string.

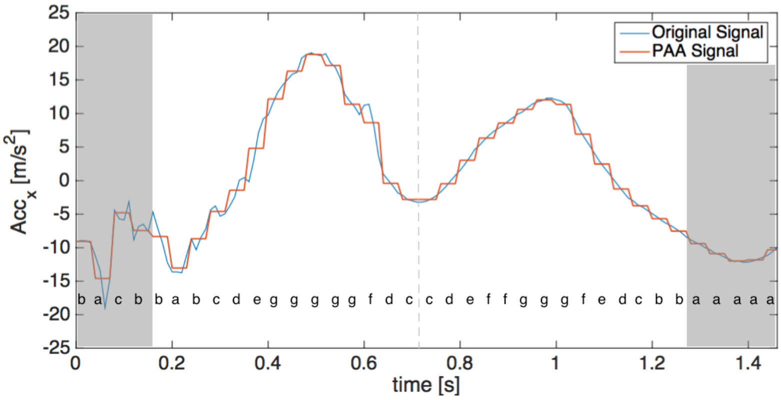


Figure 31: In the example $w = 30$, $w_1 = 28$, $\alpha = 7$. A DP Atomic Gesture with $n = 148$ that is approximated with 37 PAA frames. The word *bacbbabcdegggggfdccdeffgggfedcbbaaaaa* is the SAX string that represent the signal in Figure; from this are extracted the w_1 *central characters*. These are the characters that correspond to the lighten portion of the signal.

5 ATOMIC GESTURES CLASSIFICATION

5.1 INTRODUCTION

As previously stated, we consider 7 Athletes that perform 3 different Cross-Country Skiing techniques. Each technique represent a class of Gestures:

- DP
- KDP
- DS

For all the Atomic Gesture extracted using the Gaussian Filter, we want to be able to recognize the Atomic Gesture and so classify it in the correct class of Gesture. In order to do this, each Atomic Gesture must be converted into SAX Substring as explained in the previous Sections. Then we have to calculate the *SAX Distance* between the Atomic Gesture that we want to classify and some *reference strings (templates)*. Initially, each *template* matches one class of Gesture, so we have a template that represents DP Atomic Gesture, one KDP Atomic Gesture and another DS Atomic Gesture. Finally, we classify the Atomic Gesture in the *nearest* class of Gestures, where the metric distance is provided by SAX.
We test this procedure on the dataset where the Atomic gestures are already *labeled* as DP, KDP or DS. In this way we exactly know in which class each Atomic Gesture is included and we can calculate the *classification accuracy*.

- **Percentage of Classification accuracy:** The classification accuracy represent the number of gestures that results correctly classified in the class indicated on their *label*. The percentage will be calculated as ratio between number of gestures correctly classified and total number of gestures in exam.

In the following Sections we reported some classification results obtained with $w = 30$, $w_1 = 25$; these values allow to follow the original signal evolution, but with a reasonable dimensionality reduction effect and noise attenuation that is showed at beginning and at the end of the Atomic Gesture. Initially, $\alpha = 7$, in this way we can immediately give some impression on the particular templates choice and the logic behind the classification. However, in Section 5.3.2, we will change this value for further evaluations.

5.2 DP AND KDP AS SEPARATED CLASS

The first step for the classification is to find the three *templates*, one for each class of Gestures. Initially we are interested to classify the Atomic Gestures for one Athlete at a time. In fact, the Athletes differs for *gender* and *skills level* and thus it's thinkable that even if we consider the same class, but different Athletes, probably the gestures are still different. Moreover, in this way, it's possible to verify how precise and similar are, between them, the gestures of a single class performed by a single Athlete.
In Figure 32 we reported the classification accuracy for the seven athletes obtained with $w = 30$, $w_1 = 25$ and $\alpha = 7$; it can be noticed that only *Athlete 4* reaches an elevated percentage of classification accuracy in all his techniques performed. Many *Athletes* show an higher percentage for DS Gestures rather than their DP and KDP Gestures. This consideration means that many DP or KDP Atomic Gestures are misclassify as KDP or DP gestures, respectively and thus DP and KDP are hardly distinguishable. Furthermore, in general, we can't be satisfied by these results. For these reasons in Section 5.3 we try other ways which could improve the results.

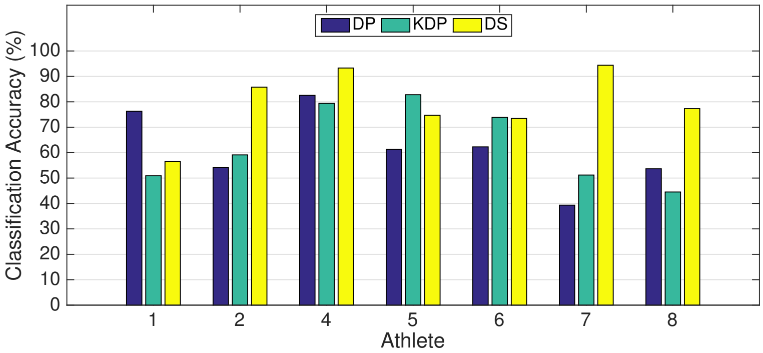


Figure 32: In this experiment: $w = 30$, $w_1 = 25$ and $\alpha = 7$. Percentage of Classification accuracy for the seven Athletes. Each class of every Athlete is represent by one template. To classify them, each athlete's gesture is confronted only with his three templates.

5.3 DP AND KDP AS SINGLE CLASS

With the experiments in Section 5.2 we noticed that a lot of DP Atomic Gestures are misclassified as KDP and vice versa. Given the device position, this misclassification seems reasonable and it may be due to a sensor-placement limitation. In fact, a sensor placed on the ankle may ensure a more realistic data collection that could represent better the "kick action", distinguishing better DP from KDP and guaranteeing also a batter classification. For this reason in the following we try to consider DP and KDP as a single class. In Section 5.3.1 we reported the results obtained with a single template that represent the single class DP/KDP while in Section 5.3.2 we reported the results obtained with two template that represent the single class DP/KDP.

5.3.1 *One template for DP/KDP*

As in Section 5.2 we want to find a number of *templates* equal to the number of class considered. Given that we want to consider DP and KDP as single class, it is necessary to find two templates, one that represent DP/KDP class and another that represent the DS class. In this way we classify an Atomic Gesture as DP if the SAX Distance between it and the DP/KDP template is lower than the SAX Distance from the DS template and vice versa.
In Figure 33 we report the percentage of Classification accuracy for the seven Athletes. It can be noticed that the percentages results not good for the majority of Athletes. With other values of α the situation doesn't improve, but we can observe that *Athlete 7* is the best Athlete in terms of Gestures similarity inside each his class.

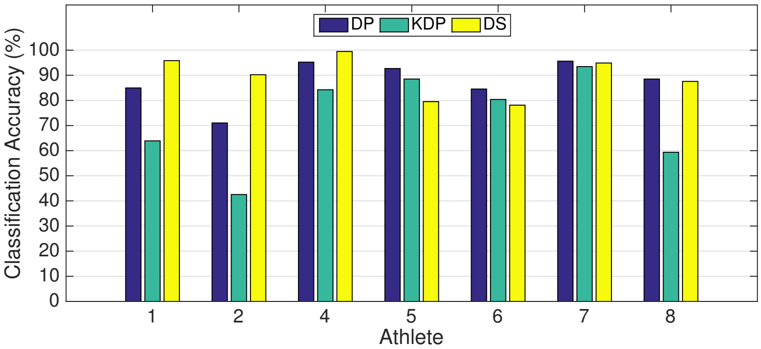


Figure 33: In this experiment: $w = 30$, $w_1 = 25$ and $\alpha = 7$. Percentage of Classification accuracy for the seven Athletes. Each class of every Athlete is represent by one template. To classify them, each athlete's gesture is confronted only with his two templates. Notice that DP classification accuracy represent the percentage of DP Atomic Gesture exactly classify into DP/KDP class, not in DP class. At the same way for KDP Atomic gesture.

5.3.2 *Two templates for DP/KDP*

In this section, we want to find three *templates*, one for DP, one for KDP and another for DS; only at a later time we consider DP and KDP as single gesture class. Specifically when we have to classify an Atomic Gesture, we evaluate which is the nearest class using the three templates. But whether the nearest class is DP or KDP, the Atomic Gesture in exam will be classified in the DP/KDP class. In this way we admit that we are unable to distinguish DP from KDP Atomic

Gesture but we reduce the misclassification percentage between DP or KDP and DS gestures. Also in these experiments we individually evaluate the Athletes so we find three templates for each Athlete. Now, we discuss about this situation varying α, with $w = 30$ and $w_1 = 25$; three examples are summarized in Table 13 and are illustrated in Figure 34, 35 and 36.

α	w	w_1	Figure ref.
4	30	25	Figure 34
7	30	25	Figure 35
10	30	25	Figure 36

Table 13: Examples of testing the classification with SAX technique varying α. At Figure ref. the *classification accuracy* correspondent results.

In Figure 34 it can be noticed that the percentages of classification accuracy are not good for the majority of Athletes. It depends from α that is too small. This consideration confirm that the alphabet size that allows to obtain the higher percentage of classification accuracy must be greater or equal than 7.

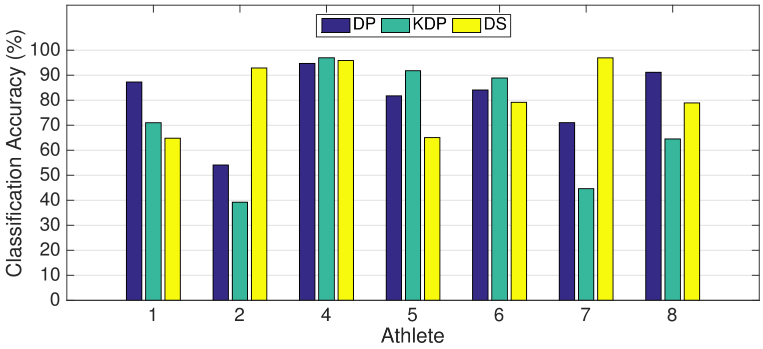


Figure 34: In this experiment: $w = 30$, $w_1 = 25$ and $\alpha = 4$. Percentage of Classification accuracy for the seven Athletes. Each class of every Athlete is represent by one template. To classify them, each athlete's gesture is confronted only with his three templates. Notice that DP classification accuracy represent the percentage of DP Atomic Gesture exactly classify into DP/KDP class, not in DP class. At the same way for KDP Atomic gesture.

In Figure 35 and 36, it can be noticed that *Athlete 4* and *Athlete 7* reach the best classification accuracy. For this reason we think that them execute the gestures always in the same manner. The *skill level* is different between the two athletes because *Athlete 4* is an Achiever but *Athlete 7* is a Recreational one. We think that the skill level influences more the movement quality rather than the similarity between Atomic Gesture

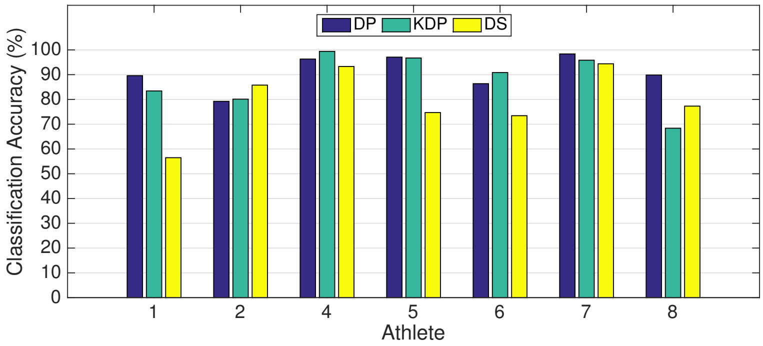


Figure 35: In this experiment: $w = 30$, $w_1 = 25$ and $\alpha = 7$. Percentage of Classification accuracy for the seven Athletes. Each class of every Athlete is represent by one template. To classify them, each athlete's gesture is confronted only with his three templates. Notice that DP classification accuracy represent the percentage of DP Atomic Gesture exactly classify into DP/KDP class, not in DP class. At the same way for KDP Atomic gesture.

included in a specific class performed by an athlete. This consideration explains why the other Achiever Athletes don't reach elevated accuracy percentage instead of what we could expect. It's also possible that a Recreational Athlete execute the Gesture always at the same manner because he doesn't personalize the technique instead an Achiever can afford to be less rigorous on the technique.

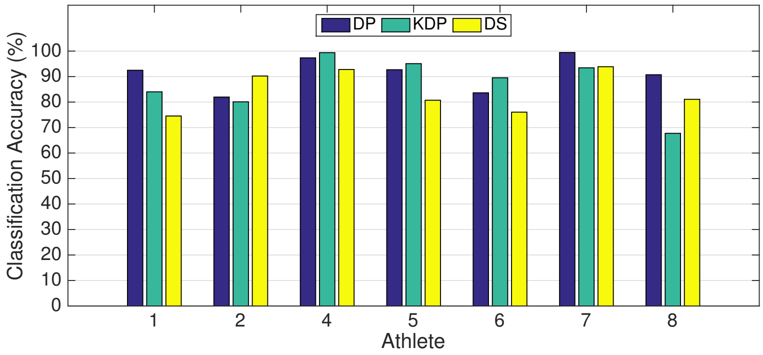


Figure 36: In this experiment: $w = 30$, $w_1 = 25$ and $\alpha = 10$. Percentage of Classification accuracy for the seven Athletes. Each class of every Athlete is represent by one template. To classify them, each athlete's gesture is confronted only with his three templates. Notice that DP classification accuracy represent the percentage of DP Atomic Gesture exactly classify into DP/KDP class, not in DP class. At the same way for KDP Atomic gesture.

Finally, comparing Figure 35 with Figure 33 the hypothesis that with *one template* to represent the class DP/KDP a lot of DP Atomic Gestures are misclassified as KDP and vice versa seems correct. In fact, in both methods the DS accuracy remains the same while the DP and KDP accuracy intensely improve. In Table 14 we reported the Sax Distance calculated between the three templates of each Athlete in the case of $\alpha = 7$. From this distance values we can't deduce interesting conclusions, but it can be noticed that, for each Athlete, the distance between DP and KDP templates is the smallest. This fact confirms that with an accelerometer placed only on the wrist, DP and KDP gestures seems indistinguishable.

Athlete	Skill Level	DP vs KDP	DP vs DS	KDP vs DS
1	R	34.12	27.04	28.62
2	A	16.03	24.75	17.36
4	A	28.80	56.75	35.76
5	A	11.68	30.82	24.70
6	R	27.52	41.98	44.11
7	R	7.78	34.18	34.71
8	R	6.95	29.13	25.43

Table 14: SAX Distance between *templates* in the case $w = 30$, $w_1 = 25$ and $\alpha = 7$. In the Table DP vs KDP stands for "SAX Distance between *DP and KDP templates*". In the same way, the others.

In Appendix A we provide the results of three experiments done with 3 templates in common between subset of Athletes in exam. In the first experiment we consider only Achiever Athletes, in the second only the Recreational ones, and in the last we consider all the Athletes. In Table 15 we summarized the results obtained in the different experiments, introducing a classification rate calculated as average between the classification accuracy of all the Athletes.

Experiment ref.	Classification Rate (%)
Figure 32	67.93
Figure 33	83.36
Figure 34	77.85
Figure 35	86.02
Figure 36	87.47

Table 15: Classification rate for the different Experiments.

5.4 FROM GESTURES TO ACTIVITY RECOGNITION

In the previous sections we have detailed the performances achieved in classifying each atomic gesture. However, it has to be reminded that we are dealing with an activity recognition problem and not with a gesture recognition one; even tough in some activity types the users may be interested in having a detailed description/classification of each gesture, in cross-country skiing performing a classification on a gesture level seems to be an excessive and superfluous target. As a consequence, in the following we will show how the proposed classification procedure performs in an activity recognition prospective.
In the following, we consider only 2 Athletes:

- An Achiever: *Athlete 4*
- A Recreational: *Athlete 7*

We simulated a training session in which, each Athlete, has to perform the three techniques DP, KDP and DS, for the same time. We want to use additional information provided from a group of subsequent Atomic gestures for the following reason: considering *continuos-repetitive activities*, it's reasonable to think that if an Athlete performs a particular techniques, he will perform it continuously for a considerable number of subsequent Atomic Gestures.

In order to consider a group of subsequent gesture, we use a *sliding window approach* that allows to evaluate the sequence the Atomic Gestures related to the training session simulation and already classified as in Section 4. In this way, we consider a *sliding window* of fixed size ℓ_{sw} that includes ℓ_{sw} subsequent Atomic Gestures at a time. Calculating the mode between the class of these ℓ_{sw} Atomic gestures, we find the the activity that represent the *window* in exam.; we memorize that the athlete has performed this activity for the period of the central Atomic Gesture of the window. After, we shift the *sliding window* of one position, considering other ℓ_{sw} Atomic gestures and so on. In Figure 37 we illustrated an example of three sliding window shifting, considering $\ell_{sw} = 3$.

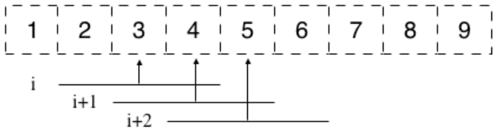


Figure 37: Examples of *sliding window* in three subsequent shifting on the Atomic Gestures sequence; $\ell_{sw} = 3$. Notice that the *arrowed lines* point to the central Atomic Gesture of each window.

We tested this approach with different sliding window size: $\ell_{sw} = 3$, $\ell_{sw} = 7$ and $\ell_{sw} = 11$. Values of ℓ_{sw} greater than 11 seem too ele-

vated because we consider reasonable that after 11 gestures of one technique, an athlete change activity. In Table 16, for each Athlete, we summarized three experiments executed considering the Atomic Gesture classification with the three class DP, KDP and DS, as in Section 5.2. In this way we show how time of Activity is recognized compared to the total time of Activity. It can be noticed that the Activity recognition works better increasing ℓ_{sw}. For *Athlete 4* the activity recognition works better than *Athlete 7*; it means that in order to obtain great results of activity recognition, we need a classification accuracy of Atomic Gestures $> 50\%$ in each class.
Instead, in Table 17, for each Athlete we summarized three experiments executed considering the Atomic Gesture classification with the two class DP/KDP and DS, as in Section 5.3.2; it can be noticed that the total activity time is recognized for $\ell_{sw} \geqslant 7$ for both the Athlete; in this case in fact, the classification accuracy of Atomic Gestures $> 50\%$ in each class.

Athlete	ℓ_{sw}	**Recognized Activity time/ Total Activity time**
4	3	10′ 41″ /11′ 41″
	7	11′ 15″ /11′ 41″
	11	11′ 19″ /11′ 41″
7	3	6′ 42"/11′ 36"
	7	6′ 59"/11′ 36"
	11	7′ 06"/11′ 36"

Table 16: Recognized Activity time on Total Activity time varying ℓ_{sw} for DP, KDP as separated class.

Athlete	ℓ_{sw}	**Recognized Activity time/ Total Activity time**
4	3	11′ 33″ /11′ 41″
	7	11′ 41″ /11′ 41″
	11	11′ 41″ /11′ 41″
7	3	11′ 29″ /11′ 36"
	7	11′ 36″ /11′ 36"
	11	11′ 36″ /11′ 36"

Table 17: Recognized Activity time on Total Activity time varying ℓ_{sw} for DP and KDP as single class.

6 NORMAL DAY ACTIVITY CLASSIFICATION

In this Chapter we want to test the algorithm detailed in the previous Chapters, with Normal Day Activity. We consider a well-known and well-documented dataset, a reduced version of the UCI Human Activity Recognition Using smartphones Dataset [72]. The dataset includes data from experiments that were carried out with a group of 30 volunteers within an age bracket of 19-48 years. They performed a protocol of activities composed of six basic activities: three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs). The experiment also included postural transitions that occurred between the static postures. These are: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand. All the participants were wearing a smartphone, Samsung Galaxy S II, on the waist during the experiment execution. In the experiments 3-axial linear acceleration and 3-axial angular velocity have been captured at a constant rate of 50 Hz using the embedded accelerometer and gyroscope of the device.

6.1 WALKING AND WALKING UP/DOWN-STAIRS

We are interested only to *Continuos-Repetitive Activity* so we consider only the data that correspond to *walking* (WLK), *walking upstairs* (WUS) and *walking downstairs* (WDS) disregarding also the postural transitions. Furthermore, each activity is composed by repetitive gestures: WLK Atomic Gestures, WUS Atomic Gestures and WDS Atomic Gestures. For simplicity, we consider only 3 *persons* randomly extracted from the 30 volunteers. We want to test the activity recognition algorithm in order to discriminate between WLK, WUS and WDS using only the only the 3-axial accelerometer data. However, for the reasons reported in Section 4.1 we have to choose only one acceleration axis. In Figure 38 an example of WDS gesture; it can be noticed that all the axes seems to contain informative content of the Gesture. However, we think that, given the sensor position, y or z axis can better discriminate between the Activities thus we test the algorithm firstly with z-axis acceleration data and then with y-axis acceleration data.

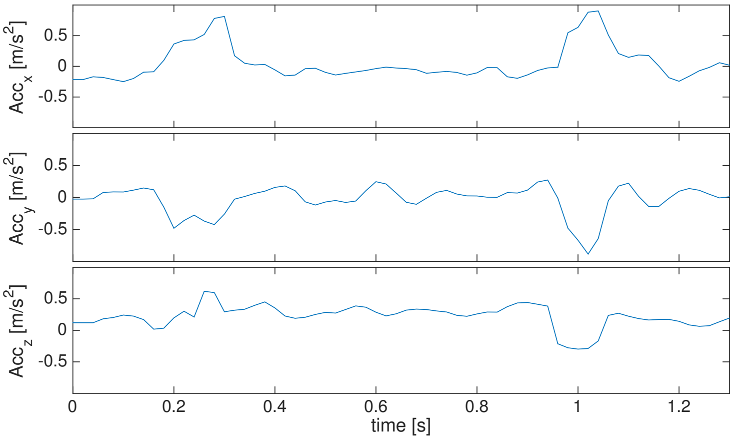


Figure 38: Example of typical WDS Atomic Gesture. The informative content of the Gesture seems to be subdivided between the three axes.

6.2 GESTURE IDENTIFICATION

Starting from the x-acceleration data stream we can obtain information about the Atomic Gesture periods using a Gaussian Filter created ad hoc. The x-acceleration data stream show a periodicity more remarked than the other axes. For this reason the filter works better on x-axis and it is easier to find the beginning and the end of the Atomic Gestures. In Figure 39 is illustrated how the Gaussian Filter allows to isolate each single gesture amidst the data stream. Each beginning and end allow to extract one Atomic Gesture signal from z-axis acceleration data stream (in the same way if we consider y-acceleration data stream).

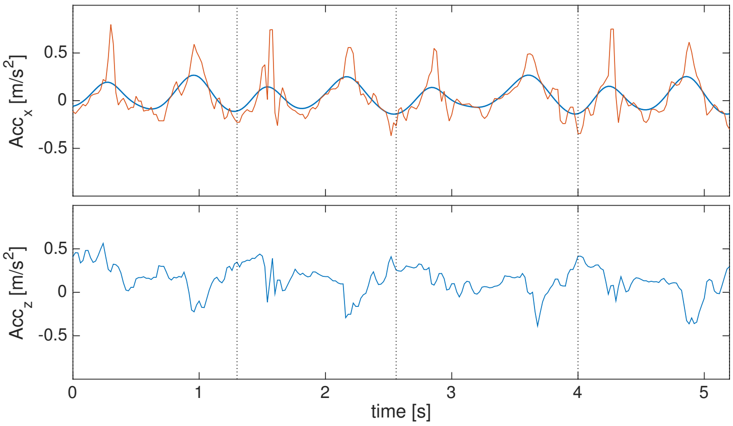


Figure 39: Filtering the x-axis signal with a Gaussian Filter we can individuate the Atomic Gestures to extract from the z-axis signal. In Figure a dotted line separates two consecutive Atomic Gestures.

6.3 ATOMIC GESTURE CLASSIFICATION

In this Section we test the recognition algorithm, resetting ad hoc the parameters w, w_1 and α. Following the considerations reported in the previous Chapters, in this situation we consider w as high as possible. The sampling rate is 50 Hz and the minimum period for an Atomic Gesture is almost 1 s, for this reason it doesn't seems reasonable to choose a value of $w > 20$. Furthermore, the signal doesn't seems affected by noise and so we can set $w_1 = 18$. The results showed in the following are obtained with $\alpha = 7$. For each person we selected three template that represent the 3 class of gestures: WLK, WUS, WDS and we evaluate the classification accuracy. In Section 6.3.1 we consider Atomic Gestures extracted from z-axis data stream while in Section 6.3.2 we consider Atomic Gestures extracted from y-axis data stream.

6.3.1 *z-Axis Atomic Gestures*

In this Section we consider the Atomic Gestures extracted from the z-axis data stream, we test the recognition algorithm evaluating the classification accuracy for the three persons in exam. In Figure 42 it can be noticed that only *Person 3* reaches an elevated percentage of classification accuracy, over 70%, in all his activities. The other *Persons* show an higher percentage for WLK Gestures rather than their WUS and WDS Gestures. This consideration can mean that many WUS or WDS Atomic Gestures are misclassify as WDS or WUS gestures, respectively. This fact is confirmed by the *confusion matrix* reported in Figure 40. Trying to solve the misclassification problem we consider WUS and WDS as single class of Gestures (WUS/DS), but using always the same three templates. In Figure 43 can be appreciated the results; it can be noticed that the percentage of classification accuracy is increased reaching a good level of activity recognition. However, in this way we can distinguish only between *walking* and *walking up or down-stairs*.

6.3.2 *y-Axis Atomic Gestures*

In this Section we consider the Atomic Gestures extracted from the y-axis data stream, we test the recognition algorithm evaluating the classification accuracy for the three persons in exam. In Figure 44 it can be noticed that an elevated percentage of WDS gestures of each Person are misclassified. As in Section 6.3.1, the *confusion matrix* in Figure 41 confirms that many WUS Atomic Gestures are misclassify WDS gestures and vice versa. In Figure 45 can be appreciated the results obtained considering WUS and WDS as single class. However, also in this case, we can't distinguish between *walking up-stairs* and *walking down-stairs*.

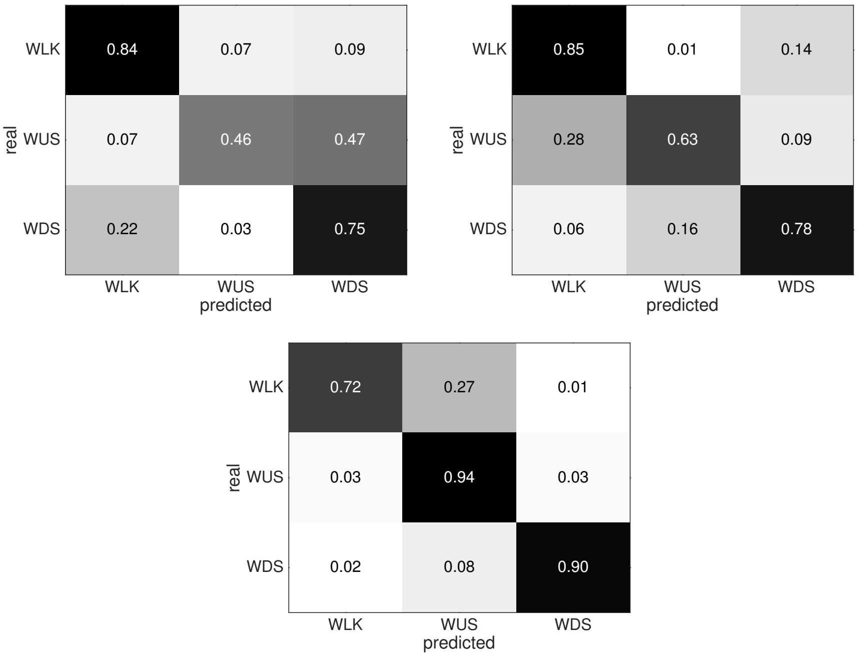


Figure 40: z-Axis Atomic Gestures. From *left* to *right*, the confusion matrix of *person 1, 2 and 3*, respectively.

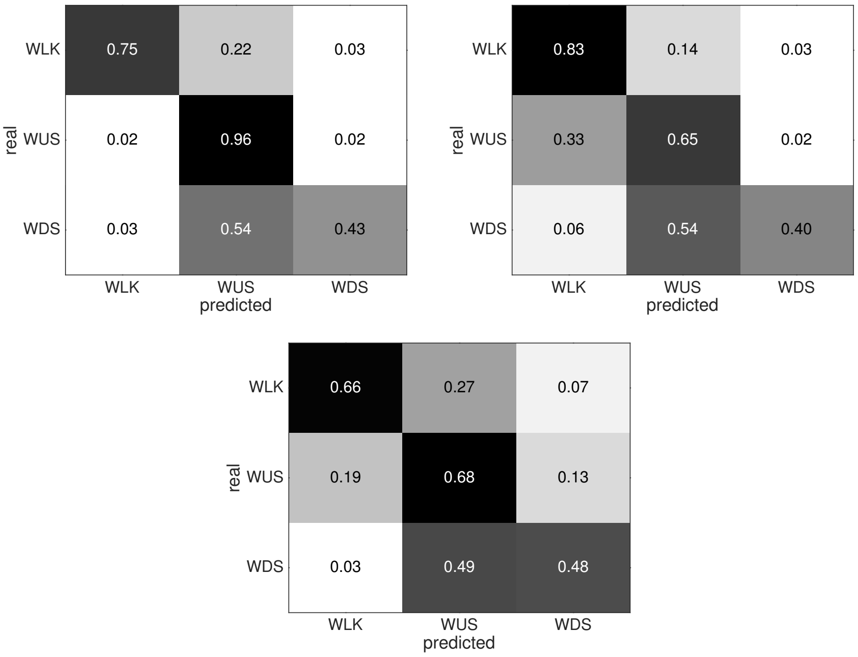


Figure 41: y-Axis Atomic Gestures. From *left* to *right*, the confusion matrix of *person 1, 2 and 3*, respectively.

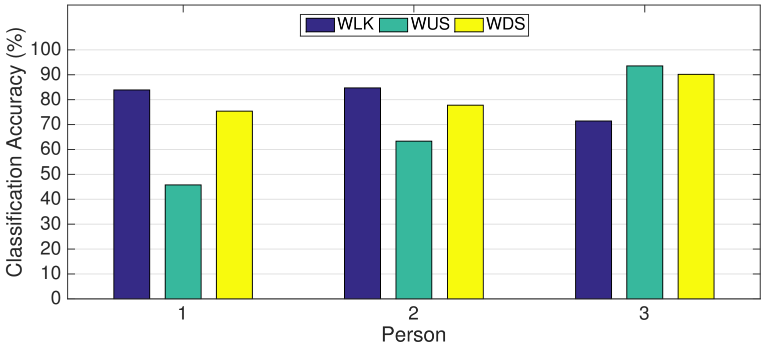


Figure 42: z-Axis Atomic Gestures. In this experiment: $w = 20$, $w_1 = 18$ and $\alpha = 7$. Percentage of Classification accuracy for the three Persons. Each class of every Persons is represent by one template. To classify them, each person's gesture is confronted only with his three templates.

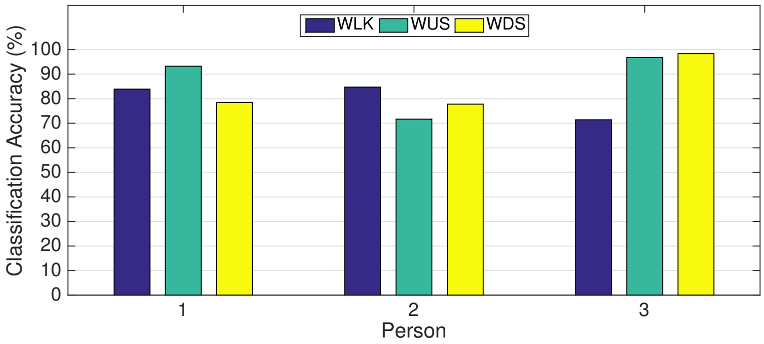


Figure 43: z-Axis Atomic Gestures. In this experiment: $w = 20$, $w_1 = 18$ and $\alpha = 7$. Percentage of Classification accuracy for the three Persons. Each class of every Persons is represent by one template. To classify them, each person's gesture is confronted only with his three templates. Notice that WUS classification accuracy represent the percentage of WUS Atomic Gesture exactly classify into WUS/DS class, not in WUS class. At the same way for WDS Atomic gesture.

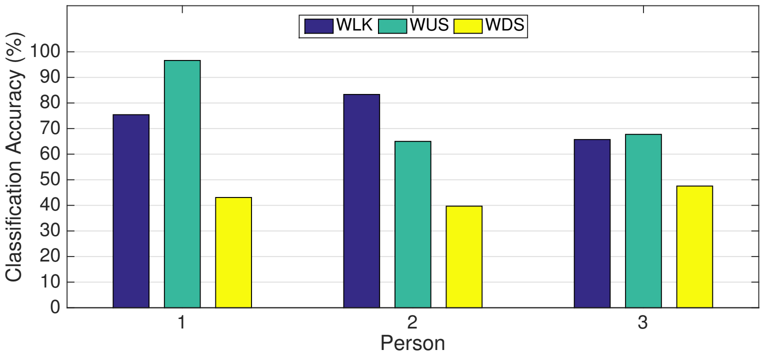


Figure 44: y-Axis Atomic Gestures. In this experiment: $w = 20$, $w_1 = 18$ and $\alpha = 7$. Percentage of Classification accuracy for the three Persons. Each class of every Persons is represent by one template. To classify them, each person's gesture is confronted only with his three templates.

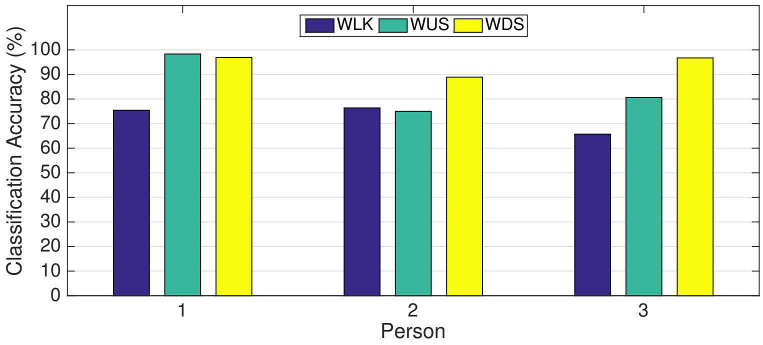


Figure 45: y-Axis Atomic Gestures. In this experiment: $w = 20$, $w_1 = 18$ and $\alpha = 7$. Percentage of Classification accuracy for the three Persons. Each class of every Persons is represent by one template. To classify them, each person's gesture is confronted only with his three templates. Notice that WUS classification accuracy represent the percentage of WUS Atomic Gesture exactly classify into WUS/DS class, not in WUS class. At the same way for WDS Atomic gesture.

6.4 general considerations

It is impossible to choose only one axis-data stream, but for apply the recognition algorithm and, more in general, the SAX technique, we had to make a choice. In Sections 6.3.1 and 6.3.2 we notice that we are not able to well distinguish between WUS and WDS. *Person 3* who shows the best classification accuracy of Atomic Gestures obtained from z-axis, is that one who obtain the worst accuracy considering y-axis data. From these results we think that a classification based only on one axis data stream is too limited.

6.5 from gestures to activity recognition

As in Section 5.4 we used the Gesture classification to solve the activity recognition problem in exam. In the following, we consider only the gesture classification based on z-axis acceleration data.
We simulated a session in which, each Person, has to walk, walk up-stairs and walk down-stairs, for the same time. We tested this approach with different sliding window size: $\ell_{sw} = 3$, $\ell_{sw} = 5$ and $\ell_{sw} = 7$. In Table 18, for each Athlete, we summarized three experiments executed considering the Atomic Gesture classification with the three class WLK, WUS and WDS, as in Section 6.3.1. In this way we show how time of Activity is recognized compared to the total time of Activity. Instead, in Table 19, for each Athlete, we summarized three experiments executed considering the Atomic Gesture classification with WUS and WDS as single class.

Person	ℓ_{sw}	**Recognized Activity time/ Total Activity time**
1	3	3′ 30″ /4′ 48″
	5	3′ 42″ /4′ 48″
	7	3′ 49″ /4′ 48″
2	3	3′ 05" /3′ 42"
	5	3′ 06" /3′ 42"
	7	3′ 09" /3′ 42"
3	3	3′ 18" /3′ 40"
	5	3′ 28" /3′ 40"
	7	3′ 29" /3′ 40"

Table 18: Recognized Activity time on Total Activity time varying ℓ_{sw} for WUS, WDS as separated class.

Person	ℓ_{sw}	**Recognized Activity time/ Total Activity time**
1	3	4′ 07″ /4′ 48″
	5	4′ 15″ /4′ 48″
	7	4′ 26″ /4′ 48″
2	3	3′ 14" /3′ 42"
	5	3′ 14" /3′ 42"
	7	3′ 15" /3′ 42"
3	3	3′ 22" /3′ 40"
	5	3′ 29" /3′ 40"
	7	3′ 29" /3′ 40"

Table 19: Recognized Activity time on Total Activity time varying ℓ_{sw} for WUS and WDS as single class.

CONCLUSION

In this thesis we focused on activity recognition, a vast topic that is widely discuss in the engineering area of interest. First, we documented about the main techniques employed in recognition problems and we retained DTW, HMM, SVM, DT/RF, BN and k-NN the most diffused. We also provided a general *Activity* description which introduce some key-concepts that characterize an Activity, such as device and sensors used to capture the data and theirs positions. Viewing the related works we understood that the sensors positions are strictly related to the specific Activities in exam; in some one cases sensor in one place is not adequate, but we need a sensors network to guarantee consistent data streams. We provided a new categorization of Activity defining three Activity Type: Continuos-Repetitive, Continuos-Spot and Isolated.

Later, we considered an activity recognition problem that concern the Cross-Country Skiing. The dataset in exam is a new collection of data obtained from the smartwatch on the wrist of 8 athletes that perform 3 different techniques of Classic Cross-Country Skiing: DP, KDP and DS that we considered as Continuos-Repetitive Activities. All the inertial sensors were embedded in the smartwatch and we didn't receive data from other devices. In our activity recognition algorithm we considered only data derived from the 3-axial accelerometer. After giving a definition of *Atomic Gesture* and *Period*, we analyzed the Atomic Gesture periods for each different techniques in exam, obtaining one reference period for DP, one for KDP and another for DS. From this first analysis we found out that KDP Atomic Gesture is slower than the others while DS Atomic Gesture is the fastest. Furthermore, we decided to overlook *Athlete 3* due to lack of data.

Our activity recognition algorithm is composed by 3 phases: Gesture Identification, SAX and Classification. In the Gesture Identification phase we extracted from the x-acceleration data stream the Atomic Gestures of each Athlete using a Gaussian Filter that allows to isolate each single gesture amidst the data stream. The algorithm core is SAX, a symbolic representation used in several fields of application that allows dimensionality reduction. We applied SAX to the Atomic Gestures setting the parameters w and α. We noticed that $w = 30$ is a value high enough to follow the original signal evolution and with a reasonable dimensionality reduction effect. However, working on Atomic Gesture it was necessary to consider the period information that differs from one to another. For this reason, we changed the common SAX technique introducing a new parameter: w_1 that represent the length of the main part of the SAX Sting. The value $w_1 = 25$

allowed to reduce the noise that is showed at beginning and at the end of the Atomic Gesture. In the classification phase, we tested the algorithm on the Atomic Gestures in the dataset in exam fixing $\alpha = 7$. We obtained poor recognition results for every Athletes, but this was reasonable because the device/sensor position don't allow to distinguishing between DP from KDP. For this reason, we considered DP and KDP as single class. With this consideration and $w = 30$, $w_1 = 25$ and $\alpha = 10$ we obtained the best results with percentages of accuracy $> 70\%$ and for two Athletes $> 90\%$. Then, we showed how the proposed classification procedure performs in an activity recognition prospective. We used a *sliding window approach*; we tested this method with different window size, obtaining great results also in the case of DP and KDP considered as separated class. We recognized $11'19''$ of Activities on a total Activity time of $11'41''$ while, if we consider DP/KDP as single class, all the activities have been correctly recognized.
In the last part, we tested the aforementioned algorithm on a well-known dataset. We considered 3 continuos-repetitive normal day Activities, played by 3 persons: Walking, Walking Up-stairs and Walking Down-stairs. In this case we had some difficulty to choose the main accelerometer axial because the informative content of the Gesture seemed to be subdivided between the three axes. For this reason we tested the algorithm firstly with z-axis acceleration data and then with y-axis acceleration data. In both cases we obtained poor recognition results. In this situations Walking Up-stairs and Walking Down-stairs seemed indistinguishable. In fact, considering this two activity as single class of gesture, with $w = 20$, $w_1 = 18$ and $\alpha = 7$ we obtained results with percentages of accuracy $> 70\%$. Applying the *sliding window approach* we obtained good results of Activity recognition.

A follow-up of this thesis can be the creation of a metrics that allows to measure the SAX Distance considering more than one acceleration axis. In this way, for example in the second dataset considered, combining the information obtained from y and z axes we may improve the recognition accuracy. Furthermore, we can can introduce an improved decision policy based on the classification performed on different axes.

APPENDIX

A

OTHER RESULTS

In This Appendix we reported other classification results omitted in the previous chapters. In Figure 46 we illustrate the results obtained evaluating separately Achiever and Recreational Athletes; we chose three templates, in common between the Athletes in exam, that represent DP, KDP and DS, respectively. However we considered DP and KDP as single class. In Figure 47 we illustrate the results obtained evaluating all the Athletes; we chose three templates, in common between all the Athletes, that represent DP, KDP and DS, respectively. However, also in this experiment, we considered DP and KDP as single class.

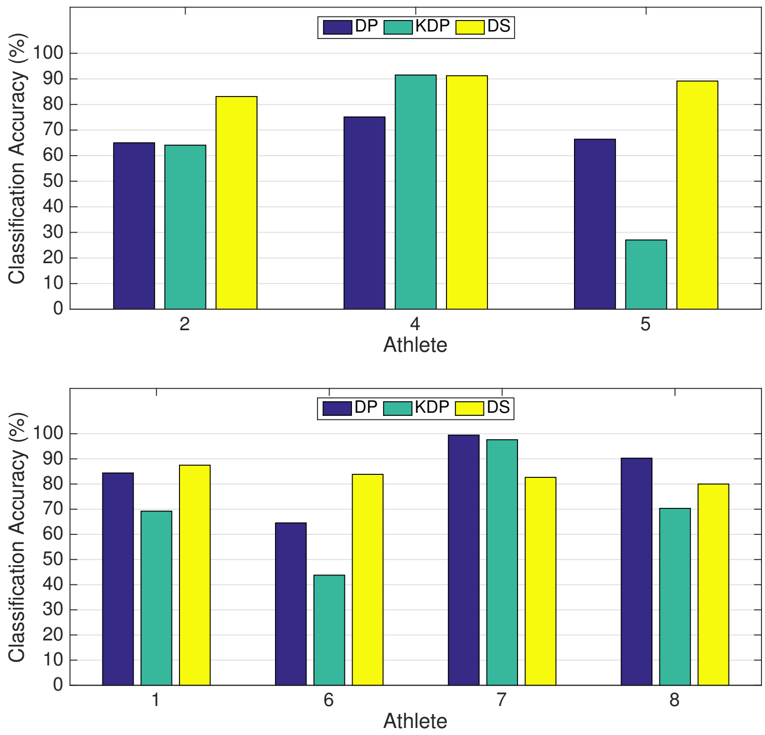


Figure 46: In these experiments: $w = 30$, $w_1 = 25$ and $\alpha = 7$. Percentage of Classification accuracy for the Achiever Athletes (*top panel*) and Recreation Athletes (*bottom panel*). Each class is represent by one template. To classify them, each athlete's gesture is confronted only with the same three templates. Notice that DP classification accuracy represent the percentage of DP Atomic Gesture exactly classify into DP/KDP class, not in DP class. At the same way for KDP Atomic gesture.

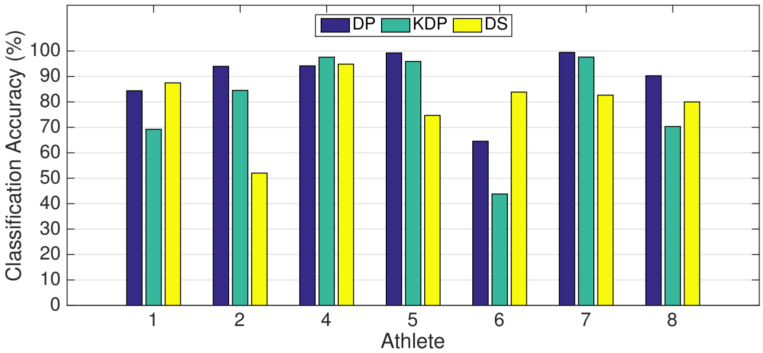


Figure 47: In this experiment: $w = 30$, $w_1 = 25$ and $\alpha = 7$. Percentage of Classification accuracy for the seven Athletes. Each class is represent by one template. To classify them, each athlete's gesture is confronted only with the same three templates. Notice that DP classification accuracy represent the percentage of DP Atomic Gesture exactly classify into DP/KDP class, not in DP class. At the same way for KDP Atomic gesture.

BIBLIOGRAPHY

[1] Ô. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2010.

[2] 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*, 2003, pp. 2–11.

[3] [Online]. Available: http://www.suunto.com

[4] [Online]. Available: http://www.garmin.com

[5] [Online]. Available: http://www.polar.com

[6] [Online]. Available: http://www.fitbit.com

[7] [Online]. Available: http://www.brainsomeness.com

[8] [Online]. Available: https://www.qlipp.com

[9] [Online]. Available: https://woosports.com/

[10] Y. Wu and T. S. Huang, "Vision-based gesture recognition: a review," *Lecture Notes in Computer Science*, vol. 1739, pp. 103–115, 1999.

[11] D. N. Monekosso and P. Remagnino, "Behavior analysis for assisted living," *Automation Science and Engineering, IEEE Transactions on*, vol. 7, pp. 879 – 886, 2010.

[12] O. Brdiczka, M. Langet, J. Maisonnasse, and J. L. Crowley, "Detecting human behavior models from multimodal observation in a smart home," *Automation Science and Engineering, IEEE Transactions on*, vol. 6, pp. 588 – 597, 2009.

[13] E. Board, "Editorial home automation as a means of independent living," *IEEE transaction on automation science and engineering*, 2008.

[14] E. D. Williams and H. S. Matthews, "Scoping the potential of monitoring and control technologies to reduce energy use in homes," *Energy and Buildings*, vol. 42, pp. 563–569, 2007.

[15] M. Blumendorf, "Building sustainable smart homes," in *Proceedings of the First International Conference on Information and Communication Technologies for Sustainability ETH Zurich*, 2013.

[16] A.-G. Paetz and E. Dütschke, "Smart homes as a means to sustainable energy consumption: A study of consumer perceptions," *Journal of Consumer Policy*, vol. 35, pp. 23–41, 2012.

[17] F. Soltani, F. Eskandari, and S. Golestan, "Developing a gesture-based game for deaf/mute people using microsoft kinect," in *Complex, Intelligent and Software Intensive Systems (CISIS), Sixth International Conference on*, 2012, pp. 491 – 495.

[18] J. Wu, G. Pan, D. Zhang, G. Qui, and S. Li, "Gesture recognition with 3d accelerometer," *Lecture Notes in Computer Science*, vol. 5585, pp. 25–38, 2009.

[19] J. Yang, E.-S. Choi, W. Chang, W.-C. Bang, S.-J. Cho, J.-K. Oh, J.-K. Cho, and D.-Y. Kim, "A novel hand gesture input device based on inertial sensing technique," in *Industrial Electronics Society, 30th Annual Conference of IEEE*, 2004, pp. 2786 – 2791.

[20] J. Kela, P. Korpipaa, J. M. S. Kallio, G. Savino, L. Jozzo, and S. D. Marca, "Accelerometer-based gesture control for a design environment," *Personal and Ubiquitous Computing*, vol. 10, pp. 285–299, 2006.

[21] Z. He, L. Jin, L. Zhen, and J. Huang, "Gesture recognition based on 3d accelerometer for cell phones interaction," in *Circuits and Systems, IEEE Asia Pacific Conference on*, 2008, pp. 217 – 220.

[22] E. Garcia-Ceja, R. Brena, and C. E. Galvan-Tejada, "Contextualized hand gesture recognition with smartphones," *Lecture Notes in Computer Science*, vol. 8495, no. 122-131, 2012.

[23] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, pp. 657–675, 2009.

[24] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 6197 – 6205, 2011.

[25] R. Xu, S. Zhou, and W. J. Li, "Mems accelerometer based nonspecific-user hand gesture recognition," *Sensors Journal, IEEE*, vol. 12, pp. 1166 – 1173, 2012.

[26] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement*, vol. 25, no. 2, 2004.

[27] C. C. Yang and Y. L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, pp. 7772–7788, 2010.

[28] A. Y. Benbasat and J. A. Paradiso, "An inertial measurement framework for gesture recognition and applications," in *Gesture and Sign Language in Human-Computer Interaction*, 2002, pp. 9–20.

[29] T. Stiefmeier, D. Roggen, and G. Tröster, "Gestures are strings: Efficient online gesture spotting and classification using string matching," in *Proceedings of the ICST 2nd international conference on Body area networks*, no. 16, 2012.

[30] D. Pruthi, A. Jain, K. M. Jatavallabhula, R. Nalwaya, and P. Teja, "Maxxyt: An autonomous wearable device for real-time tracking of a wide range of exercises," in *17th UKSIM-AMSS International Conference on Modelling and Simulation*, 2015.

[31] Y. Zhang, W. Liang, J. Tan, Y. Li, and Z. Zeng, "Pca and hmm based arm gesture recognition using inertial measurement unit," in *Proceedings of the 8th International Conference on Body Area Networks*, 2013, pp. 193–196.

[32] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, pp. 20 – 26, 2008.

[33] K. Tsukadaa and M. Yasumura, "Ubi-finger: Gesture input device for mobile use," in *Ubicomp, Informal Companion Proceedings*, 2002.

[34] A. A. Chaaraoui, J. R. Padilla-Lopez, F. J. Ferrandez-Pastor, M. Nieto-Hidalgo, and F. Florez-Revuelta, "A vision-based system for intelligent monitoring: Human behaviour analysis and privacy by context," *Sensors*, vol. 14, pp. 8895–8925, 2014.

[35] L. Chan, C.-H. Hsieh, Y.-L. Chen, S. Yang, D.-Y. Huang, R.-H. Liang, and B.-Y. Chen, "Cyclops: Wearable and single-piece full-body gesture input devices," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 3001–3009.

[36] M. Burke and J. Lasenby, "Pantomimic gestures for human–robot interaction," *Robotics, IEEE Transactions on*, vol. 31, pp. 1225 – 1237, 2015.

[37] A. Gardner, C. A. Duncan, J. Kanno, and R. Selmic, "3d hand posture recognition from small unlabeled point sets," in *Systems, Man and Cybernetics (SMC), IEEE International Conference on*, 2014, pp. 164 – 169.

[38] D. Sturman and D. Zeltzer, "A survey of glove-based input," *Computer Graphics and Applications, IEEE*, vol. 14, pp. 30 – 39, 1994.

[39] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, pp. 461 – 482, 2008.

[40] J. Liu, Z. Pan, and X. Li, "An accelerometer-based gesture recognition algorithm and its application for 3d interaction," *Computer Science and Information Systems*, vol. 7, pp. 177–188, 2010.

[41] X. Long, B. Yin, and R. M. Aarts, "Single accelerometer-based daily physical activity classification," in *Engineering in Medicine and Biology Society, Annual International Conference of the IEEE*, 2009, pp. 6107 – 6110.

[42] E. S. Sazonov, G. Fulk, J. Hill, Y. Schutz, and R. Browning, "Monitoring of posture allocations and activities by a shoe-based wearable sensor," *Biomedical Engineering, IEEE Transactions on*, vol. 58, pp. 983 – 990, 2011.

[43] A. Cenedese, G. A. Susto, G. Belgioioso, G. I. Cirillo, and F. Fraccaroli, "Machine learning gesture classification from inertial measurements for home automation," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, pp. 1200 – 1210, 2014.

[44] D. Morris, T. S. Saponas, A. Guillory, and I. Kelner, "Recofit: Using a wearable sensor to find, recognize, and count repetitive exercises," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 3225–3234.

[45] H. Ghasemzadeh, V. Loseu, E. Guenterberg, and R. Jafari, "Sport training using body sensor networks: A statistical approach to measure wrist rotation for golf swing," in *Proceedings of the Fourth International Conference on Body Area Networks*, 2009.

[46] E. A. Heinz, K. S. Kunze, M. Gruber, D. Bannach, and P. Lukowicz, "Using wearable sensors for real-time recognition tasks in games of martial arts – an initial experiment," in *Computational Intelligence and Games, IEEE Symposium on*, 2006, pp. 98 – 102.

[47] A. Ahmadi, E. Mitchell, F. Destelle, M. Gowing, N. E. O'Connor, C. Richter, and K. Moran, "Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors," in *Wearable and Implantable Body Sensor Networks (BSN), 11th International Conference on.*, 2014, pp. 98–103.

[48] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Data Mining, Proceedings IEEE International Conference on*, 2001, pp. 289 – 296.

[49] E. Guenterberg, S. Ostadabbas, H. Ghasemzadeh, and R. Jafari, "An automatic segmentation technique in body sensor networks based on signal energy," in *Proceedings of the Fourth International Conference on Body Area Networks*, 2009.

[50] A. Parate, M.-C. Chiu, C. Chadowitz, and D. Ganesan, "Risq: Recognizing smoking gestures with inertial sensors on a wristband," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 2014.

[51] Müller and Meinard, *Information Retrieval for Music and Motion*. Springer, 2007.

[52] L. R.Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257 – 286, 1989.

[53] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, 4th ed. Springer, 2014.

[54] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[55] M. E. Tipping, "Sparse bayesian learning and relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[56] C. C. Aggarwal, *Data Mining: The Textbook*. Springer, 2015.

[57] J. Shieh and E. Keogh, "isax: Indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 623–631.

[58] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, "Multi-layered gesture recognition with kinect," *Journal of Machine Learning Research*, vol. 16, pp. 227–254, 2015.

[59] V. Pitsikalis, A. Katsamanis, S. Theodorakis, and P. Maragos, "Multimodal gesture recognition via multiple hypotheses rescoring," *Journal of Machine Learning Research*, vol. 16, pp. 255–284, 2015.

[60] S.-J. Cho, E. Choi, W.-C. Bang, J. Yang, J. Sohn, D. Y. Kim, Y.-B. Lee, and S. Kim, "Two-stage recognition of raw acceleration signals for 3-d gesture-understanding cell phones," in *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[61] C. Tran and M. M. Trivedi, "3-d posture and gesture recognition for interactivity in smart spaces," *Industrial Informatics, IEEE Transactions on*, vol. 8, pp. 178 – 187, 2012.

[62] M. Gowing, A. Ahmadi, F. Destelle, D. Monaghan, N. O'Connor, and K. Moran, "Kinect vs. low-cost inertial sensing for gesture recognition," *Lecture Notes in Computer Science*, vol. 8325, pp. 484–495, 2014.

[63] I. Androulakis, "New approaches for representing, analyzing and visualizing complex kinetic mechanisms," in *38th European Symposium of the Working Party on Computer Aided Process Engineering*, 2005, pp. 235–240.

[64] S. Kasetty, C. Stafford, G. P. Walker, X. Wang, and E. Keogh, "Real-time classification of streaming sensor data," in *Tools with Artificial Intelligence, 20th IEEE International Conference on*, 2008, pp. 149 – 156.

[65] L. Yi, B. De-pei, and Y. Ze-hong, "Research of a new symbolic approximation method-sax in time series mining," *Computer Engineering and Applications*, vol. 42, no. 27, 2006.

[66] F. Carbone, "Ricerca di anomalie su serie temporali di temperature basata su rappresentazione simbolica," Master's thesis, Università di Padova, 2014.

[67] M. Okabe, T. Miwa, and K. Umemura, "Anomaly detection in network traffic based on string analysis," in *IC2006*, 2006.

[68] E. Keogh, J. Lin, and A. Fu, "Hot sax: Efficiently finding the most unusual time series subsequence," in *Data Mining, Fifth IEEE International Conference on*, 2005.

[69] E. Keogh, K. Chakrabarti, M.Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," in *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, 2001, pp. 151–162.

[70] B. K. Yi. and C. Faloutsos, "Fast time sequence indexing for arbitrary lp norms," in *Proceedings of 26th International Conference on Very Large Data Bases*, 2000.

[71] M. Terzi, G. Susto, and A. Cenedese, "Technical report on activity recognition," *automatica.it*, 2015.

[72] [Online]. Available: https://sites.google.com/site/harsmartlab/

[73] T. Amr, "Survey on time-series data classification," in *TSDM*, 2012.

[74] S. Lenser, M. V. S. Lenser, and M. Veloso, "Non-parametric time series classification," in *Robotics and Automation, ICRA Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 3918 – 3923.