A quantitative comparison of Overlapping and Non-overlapping sliding windows effects for human activity recognition using inertial sensors

Akbar Dehghani

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Computer Science) at

Concordia University

Montréal, Québec, Canada

December 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Akbar Dehghani**

Entitled: **A quantitative comparison of Overlapping and Non-overlapping sliding windows effects for human activity recognition using inertial sensors**

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

Master of Applied Science (Computer Science)

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

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Denis Pankratov

______________________________ External Examiner
Dr. Adam Krzyzak

______________________________ Examiner
Dr. Essam Mansour

______________________________ Supervisor
Dr. Emad Shihab

______________________________ Co-supervisor
Dr. Tristan Glatard

Approved by ______________________________
Lata Narayanan, Chair
Department of Computer Science and Software Engineering

____________ 2019

____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

A quantitative comparison of Overlapping and Non-overlapping sliding windows effects
for human activity recognition using inertial sensors

Akbar Dehghani

The sliding window technique is widely used to segment inertial sensor signals, i.e., accelerometers and gyroscopes, for activity recognition. In this technique, the sensor signals are partitioned into fix-sized time windows which can be of two types: (1) non-overlapping windows, in which time windows do not intersect, and (2) overlapping windows, in which they do. There is a generalized idea about the positive impact of using overlapping sliding windows on the performance of recognition systems in Human Activity Recognition. In this thesis, we analyze the impact of overlapping sliding windows on the performance of Human Activity Recognition systems with different evaluation techniques, namely subject-dependent cross validation and subject-independent cross validation. Our results show that the performance improvements regarding to overlapping windowing reported in the literature seem to be associated with the underlying limitations of subject-dependent cross validation. Furthermore, we do not observe any performance gain from the use of such technique in conjunction with subject-independent cross validation. We conclude that when using subject-independent cross validation, non-overlapping sliding windows reach the same performance as sliding windows. This result has significant implications on the resource usage for training the human activity recognition systems.

Acknowledgments

First and foremost, I would like to thank God for blessing me with this opportunity.

I would like to take this opportunity to express my gratitude and appreciation to my supervisors, Dr. Emad Shihab and Dr. Tristan Glatard for their endless support and commitment during this journey. This work would have not been possible without their unparalleled motivation and immense knowledge. I would like to wish my supervisors and their esteemed families all the best in life.

My appreciation extends to my colleagues at /bin and DAS labs, especially Ali Salari, Soudabeh Barghi, Valerie Hayot-Sasson, Martin Khannouz, and Bhupinder Kaur.

I would like to thank Dr.Omid Sarbishei for his great contribution to publishing this work on the Sensors journal.

Most importantly, I would like to thank my parents and my brother, Behdad, who have always supported me and have inspired me to move forward at most difficult times. They are far and yet very near.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter includes an overview of the thesis, along with a description of the challenges. A summary is also given on contributions that are brought forth into this thesis.

1.1 Research Problem and Scope

The tremendous advances in microelectronics and computer systems during the past decades enable wearable and mobile devices with remarkable abilities which are transforming society at a rapid pace. Such abilities, including high computational power, low cost, and small size have provided a context in which people interact with such devices as inseparable parts of their life, which results in producing huge volumes of data. There exists an active research area named *Ubiquitous Sensing* with the main goal of extracting knowledge from the such data (Perez, Labrador, & Barbeau, 2010). In particular, thanks to the importance of comprehension of human demands and necessities, *human activity recognition* (HAR) has become a task of high interest among researchers. Such understanding has many applications in different fields including security, virtual reality, sports training, and especially health care. For instance, this information has been used to detect anomalous behaviors such as falls (Bianchi, Redmond, Narayanan, Cerutti, & Lovell, 2010; Luštrek & Kaluža, 2009; Tamura, Yoshimura, Sekine, Uchida, & Tanaka, 2009) and track movement-related conditions in seniors (K.-Y. Chen, Harniss, Patel, & Johnson, 2014). Another example is recognizing human activities to promote healthier lifestyles by encouraging do exercise (Consolvo et al., 2008; Lin, Mamykina, Lindtner, Delajoux, & Strub, 2006) and preventing unhealthy habits such as tobacco use or unwholesome food (E. Sazonov, Metcalfe, Lopez-Meyer, & Tiffany, 2011; E. S. Sazonov et al., 2009).

The recognition of human activities has been approached in two different ways, namely using external and wearable sensors (Lara & Labrador, 2012). In the external scenario, the sensors are fixed in prearranged points and the recognition depends on the interaction of users with the sensors and also being accessible to them. As for wearable sensing, however, the sensors are attached to the user's body and consequently, recognition is invariant to user interaction and position respect to the sensors. In general, wearable sensing has been proven to be the prevalent technology in HAR (Banos, Galvez, Damas, Pomares, & Rojas, 2014);

A classic example of using external sensors to recognize human activities is intelligent homes (Kasteren, Englebienne, & Kröse, 2010; Sarkar, Lee, Lee, et al., 2011; Tolstikov et al., 2011; Yang, Lee, & Choi, 2011). In these systems, sensors are placed in target objects (e.g. stove, fridge, washing machine, etc) and user's activities are recognized during interaction with them. Another example of external sensing is using cameras to recognize human activities (Ahad, Tan, Kim, & Ishikawa, 2008; Candamo, Shreve, Goldgof, Sapper, & Kasturi, 2009; Joseph et al., 2010; Turaga, Chellappa, Subrahmanian, & Udrea, 2008) which is especially suitable for security and gaming (Shotton et al., 2011). In general, systems with external sensors are able to recognize fairly complex activities such as eating, taking a shower, washing dishes, etc (Lara & Labrador, 2012). However, such systems have several issues including privacy, pervasiveness, complexity, and maintenance cost (Lara & Labrador, 2012) which motivate to use of wearable sensors in HAR.

Wearable fitness devices such as the Nike Fuel Band and Fit-Bit Flex are good examples for wearable sensing. These devices, which can be used during exercise, track their wearers physical activities (e.g. number of steps taken) and caloric expenditure. Users may use such information to set their own fitness goals and monitor their progression. In general, the employed sensors in wearable devices can be categorized into three groups: (1) sensors to measure the user's movement (e.g. accelerometer, gyroscope, GPS, etc), (2) sensors to measure the environmental variables (e.g. temperature and humidity), and (3) sensors to measure the physiological signals (e.g., heart rate and electrocardiogram).

Most HAR systems use an Activity Recognition Process (ARP) to detect activities. These systems usually consist of one or more sensors attached to different parts of a person's body that provide diverse streams of sensor data. Such data streams, subsequently, are segmented into several time windows with specific length and from which feature vectors are extracted and fed to a classifier.

1.2 Motivation

Segmentation in time windows is a critical process in ARP, often implemented with sliding windows (Banos et al., 2014; Janidarmian, Roshan Fekr, Radecka, & Zilic, 2017) that can be of two types: (1) non-overlapping windows, in which time windows do not intersect, and (2) overlapping windows, in which they do (Lara & Labrador, 2012). Both overlapping and non-overlapping windows are commonly used in the literature. For instance, in Recofit (Morris, Saponas, Guillory, & Kelner, 2014), sensor signals are windowed into 5-second overlapping windows sliding at 200ms. Another example is the work in (Banos et al., 2014), that uses non-overlapping sliding windows to partition sensor signals.

Several works (Janidarmian, Radecka, & Zilic, 2014; Keogh, Chu, Hart, & Pazzani, 2001) have shown that using overlapping sliding windows instead of non-overlapping ones improves the accuracy of the recognition systems. However, the amount of such improvement and its sources in HAR remain unclear. This study addresses this question: based on a detailed, quantitative analysis of multiple datasets, we explain why and by how much overlapping windows affect the performance of ARP. We report and discuss the general and per activity impacts of these two methods considering two cross validation (CV) techniques namely subject-dependent CV and subject-independent CV.

1.3 Thesis Contributions

The major contributions of the thesis are as follows:

- An in-depth investigation of how HAR system performance is impacted by overlapping and non-overlapping sliding windows.

- An investigation of the main reasons of performance improvement by overlapping sliding windows in HAR systems
- A set of publicly available scripts[1] to help the research community further shed light on the important topic of choosing the types of sliding windows in HAR.

1.4 Thesis Overview

Chapter 2 discusses the background of HAR using wearable devices. This Chapter is divided into two parts. The first part describes the different stages of ARP and the commonly used techniques in each stage. Moreover, this part explains and compares diverse types of system evaluation techniques in HAR systems. The second part of Chapter 2 reviews the segmentation process using the sliding window technique in previous works in HAR fields. Chapter 3 explains the used datasets in this study and our ARP setting including utilized window sizes, feature sets, and classifiers. Furthermore, the used classifiers in this work are explained precisely in this chapter. Chapter 4 presents the results and conclusions of five different experiments were conducted to investigate the general and per activity impacts of overlapping and non-overlapping sliding windows on HAR systems. For each experiment, setting, results, and conclusion are presented. Chapter 5 discusses and compares the results of different experiments. Finally, we conclude the thesis by summarizing our work and findings and offering remarks on possible future works. All of the chapters of this thesis except for some small parts of Chapter 1 and Chapter 2 have been published in Sensors journal (Dehghani, Sarbishei, Glatard, & Shihab, 2019).

[1] http://www.github.com/big-data-lab-team/paper-generalizability-window-size

Chapter 2

Human Activity Recognition using Wearable sensors

2.1 Background

In this section, we provide an overview of a typical activity recognition process (ARP) explain and compare different sliding windows techniques and describe common evaluation methods in HAR.

2.1.1 Activity Recognition Process

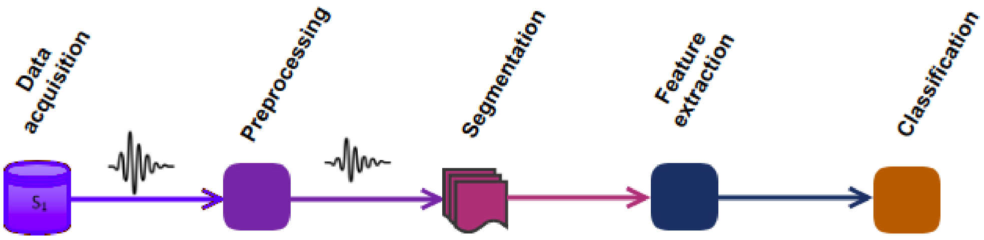


Figure 2.1: Human activity recognition process

ARP is composed of a sequence of signal processing, pattern recognition, and machine learning techniques (Bulling, Blanke, & Schiele, 2014). It mainly consists of 5 steps, shown in Figure 2.1 and explained hereafter.

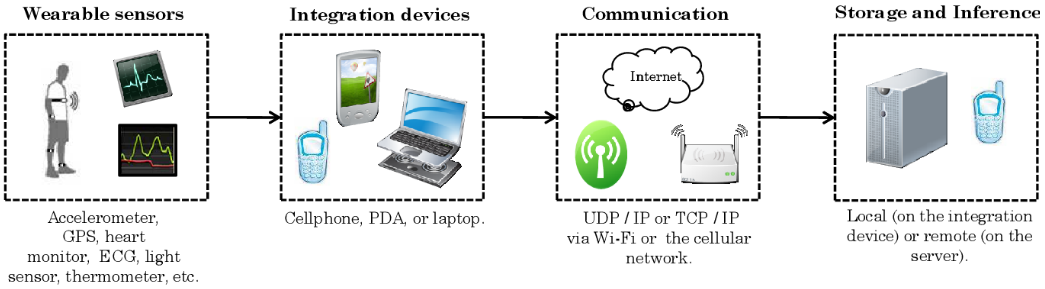

Wearable sensors
Integration devices
Communication
Storage and Inference
Internet
Accelerometer, GPS, heart monitor, ECG, light sensor, thermometer, etc.
Cellphone, PDA, or laptop.
UDP / IP or TCP / IP via Wi-Fi or the cellular network.
Local (on the integration device) or remote (on the server).

Figure 2.2: Generic data acquisition architecture for HAR (extracted from (Lara & Labrador, 2012)).

Data acquisition

Several sensors are attached to different body parts. They mostly acquire 3D acceleration, gyroscopic and magnetic field measurements, etc, which measure different attributes such as motion (Iglesias, Cano, Bernardos, & Casar, 2011), location (Choujaa & Dulay, 2008), temperature (Parkka et al., 2006), and ECG (Jatoba, Grossmann, Kunze, Ottenbacher, & Stork, 2008). Sensors discretize signals at a given frequency, typically 50Hz for daily activities or 200Hz for fast sports, and transmit data through UDP/IP or TCP/IP protocols to integration devices (ID) to be preprocessed (Lara & Labrador, 2012). IDs can be different devices including cellphones, PDAs, laptops or a customized embedded systems (Lara & Labrador, 2012). Figure 2.2 shows a generic architecture of data acquisition in HAR. It should be noted that all of the mentioned components are not necessarily implemented in every HAR systems.

Preprocessing

Data points coming from sensors may include artifacts of various origins such as electronic fluctuations, sensor malfunctions, and physical activities (Bulling et al., 2014). To eliminate such artifacts, filtering techniques are commonly applied, such as the Butterworth low-pass filter, which flats the coming signals as much as possible through rolling off down the higher frequencies beyond the cut-off point to zero (Morris et al., 2014; Najafi et al., 2003; Selles, Formanoy, Bussmann, Janssens, & Stam, 2005). In any case, preprocessing techniques need to preserve the signal characteristics that carry relevant information about the activities and consequently, they should be used with care as they may remove valuable information from the signals.

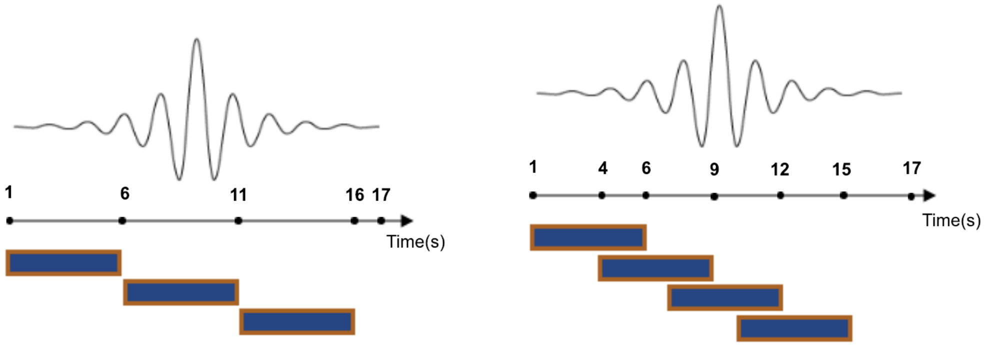
Segmentation

In this stage, the coming signals from sensors are partitioned into time windows labeled from the most frequent activity in the window. There are several ways to segment the sensor signals in HAR field which can be categorized into three groups, namely activity-defined windows, event-defined windows and sliding windows (Banos et al., 2014).

Activity-defined windows.In this technique, signals are partitioned based on the detection of activity changes, so the start and end points of each activity should be determined. Since the times of activities can be different, the window sizes are not fixed. In the literature, several methods have been used to identify the activity-transition points. For instance, Nyan, Tay, Seah, and Sitoh (2006); Sekine, Tamura, Togawa, and Fukui (2000) suggest a model based on the variations in the frequency characteristics to identify activity-transition points. Another example is the work in (Yoshizawa, Takasaki, & Ohmura, 2013) which proposes a heuristic method to separate static actions from dynamic ones. In the simpler scenario, the activity-transition points can be identified by user feedback (Dernbach, Das, Krishnan, Thomas, & Cook, 2012; Figo, Diniz, Ferreira, & Cardoso, 2010; He & Jin, 2009; Lester, Choudhury, & Borriello, 2006).

Event-defined windows. Some activities such as meal preparation could be better recognized as a sequence of actions performed in a certain order. However, such activities are irregular and the order of actions for a specific one can be different and under such circumstances, the identification of specific events is particularly advised. The goal of event-defined windows is locating specific events in the signals to be used for data segmentation. In particular, this method is widely used in gait analysis (Banos et al., 2014). In this case, the models aim to detect heel strikes (the initial floor contact) and toe-offs (the end of floor contact) events. In the literature, several methods have been utilized to identify such events including analyzing the foot's linear acceleration (Aminian et al., 1999;

Selles et al., 2005), Foot (Aminian, Najafi, Büla, Leyvraz, & Robert, 2002) and shank (Jasiewicz et al., 2006) sagittal angular velocity, using clustering models such as Gaussian mixture (Aung et al., 2013), using external mechanisms like stopwatch to register the first crossing time of hind foot and lead foot from a given start and end lines respectively (Dobkin, Xu, Batalin, Thomas, & Kaiser, 2011), etc. Identifying such events, subsequently can be used to recognize activities. For example, Benocci et al. (2010) use a model to recognize walking by identifying the gait cycle on a single foot tagged through a heel strike event. Like activity-defined windows, the window sizes are not fixed since the events may not be uniformly distributed in time.



(a) Non-overlapping (b) Overlapping-2 s sharing

Figure 2.3: 5-second sliding windows.

Sliding windows. The sliding window approach is the most widely used method in the segmentation step of ARP due to its implementational simplicity and lack of preprocessing (Banos et al., 2014). In this approach, the sensor signals are split into windows of fixed size. If there is overlap between adjacent windows, this technique is known as overlapping sliding window, and if not, it is called non-overlapping windows technique. Figure 2.3 illustrates the non-overlapping and overlapping windowing techniques.

Formal definition- Mathematically, a sliding window process can be defined as follows. Assume a stream of data values $x_i \in R$ at times t_i $(i \in N)$. For simplicity, we assume that $t_0 = 0$ and the sampling period remains constant at ΔT, i.e.,

$$\forall i \in N, t_{i+1} - t_i = \Delta T.$$

A fixed length sliding window splits the data stream into individual segments, where each segment consists of n $(n \in N, n > 1)$ samples. Consequently, the window size T in seconds is computed as follows:

$T = (n - 1)\Delta T,$

where ΔT is the sampling period. We also denote $p \in \{0, 1, 2, ..., n - 1\}$ as the number of samples that are within the overlapping period between two consecutive windows, where p = 0 refers to the scenario with non-overlapping windows. Next, the overlapping period between two consecutive windows in seconds, i.e., OP, can be computed as follows:

$OP = p\Delta T.$

Many research articles define the overlapping period as a percentage of the overall window length, e.g., 80% overlapping windows. The overlapping period in percentage can also be found as follows:

$OP(\%) = \frac{p}{n}$

Finally, we can express each window/segment $S_k(k \in N)$ as a set of data values x_i as follows:

$S_K = \{x_{k(n-p)}, x_{k(n-p)+1}, ..., x_{k(n-p)+n-1}\}, (k \in N)$

There is a generalized idea that using overlapping sliding windows increases the performance of classifiers in HAR (Janidarmian et al., 2014), since they involve more data points, and unlike the non-overlapping windows, they are not prone to missing important events (Coggeshall & Wu, 2005), particularly within activity transition periods. While these assumptions are generally true, we will show later with our detailed experiments that non-overlapping windows overall deliver comparable recognition accuracy, while majorly reducing the required training computations and memory usage. The number of data points in a time window, a.k.a the window size, heavily impacts the performance of the model (Banos et al., 2014; Bulling et al., 2014). Finding the optimal window size depends on the specific requirements of the HAR system. For instance, the number of activities for which system is devised or special recognition time (Banos et al., 2014). However, in any case, the window size should be properly selected in such a way that each window contains enough samples (at least one cycle of an activity) to be differentiable from similar movements (Janidarmian et al., 2017). The current method to select the window size is empirical (Bulling et al., 2014). We apply ARP with different window sizes which mostly selected from the values used in previous works, and choose the one which maximizes the performance of the recognition system. This process can be very time consuming due to the fact that there is no prior knowledge about the optimal window size and the entire space should be searched in an uninformed way. Table 2.1 shows an extensive review of used window size in previous studies.

Window size range (s)	publications
0 - 1	Pirttikangas, Fujinami, and Nakajima (2006), Stikic, Huynh, Van Laerhoven, and Schiele (2008), Marx (2012), Huynh and Schiele (2005), Kern, Schiele, and Schmidt (2003), Maurer, Smailagic, Siewiorek, and Deisher (2006a), Suutala, Pirttikangas, and Röning (2007), Amft and Tröster (2008), Han, Kang, and Kim (2010), J.-H. Wang, Ding, Chen, and Chen (2012)
1 - 2	Pirttikangas et al. (2006), Stikic et al. (2008), Huynh and Schiele (2005), Sun, Zhang, Li, Guo, and Li (2010), Gjoreski and Gams (2011)
2 - 3	Mannini, Intille, Rosenberger, Sabatini, and Haskell (2013), Stikic et al. (2008), Preece, Goulermas, Kenney, and Howard (2008), Mantyjarvi, Himberg, and Seppanen (2001), Huynh and Schiele (2005), N. Wang, Ambikairajah, Lovell, and Celler (2007), Khan, Lee, Lee, and Kim (2010), Sun et al. (2010),Nam and Park (2013b),Nam and Park (2013a)
3 - 4	Sun et al. (2010)
4 - 5	Mannini et al. (2013), Stikic et al. (2008), Huynh and Schiele (2005),Parkka et al. (2006), Sun et al. (2010)
5 - 6	Ravi, Dandekar, Mysore, and Littman (2005), Altun and Barshan (2010), Sun et al. (2010), Atallah, Lo, King, and Yang (2011), Lee and Cho (2011)
6 - 7	Bao and Intille (2004a), Huynh, Blanke, and Schiele (2007), Sun et al. (2010), Jiang, Shang, Wang, Li, and Wang (2011)
7+	Mannini et al. (2013), Stikic et al. (2008), Krause, Siewiorek, Smailagic, and Farringdon (2003), Parkka et al. (2006), Kwapisz, Weiss, and Moore (2011), Siirtola and Röning (2012), Hemalatha and Vaidehi (2013), Zheng, Wong, Guan, and Trost (2013)

Table 2.1: Distribution of the activity recognition research studies based on the utilized window size

Group	Methods	Publications
Time domain	Mean, Standard deviation, Variance, Interquartile range, Mean absolute deviation, Correlation between axes, Kurtosis, Root Mean Square, Averaged derivatives, Skewness, Zero Crossing Rate, Mean Crossing Rate, Pairwise Correlation, Spectral Entropy	Banos et al. (2014), Morris et al. (2014), Parkka et al. (2006), Tapia et al. (2007), Kao, Lin, and Wang (2009), Maurer et al. (2006a), Zhang and Sawchuk (2011)
Frequency domain	Fourier Transform, Discrete Cosine Transform	Bao and Intille (2004a), Morris et al. (2014), Y.-P. Chen, Yang, Liou, Lee, and Wang (2008), Altun and Barshan (2010)
Others	Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Autoregresive Model (AR), HAAR filters	Morris et al. (2014), Altun and Barshan (2010), He and Jin (2009), Y.-P. Chen et al. (2008), Hanai, Nishimura, and Kuroda (2009)

Table 2.2: Summery of common used features in HAR publications

Feature extraction

The main reason for applying feature extraction on each time window is that it is almost impossible for two given time windows to be exactly identical even if they come from the same subject doing the same activity (Lara & Labrador, 2012). In the other words, through that we can filter out relevant information and obtain a quantitative measure of each time window which can further be used to compare with other time windows. In general, feature extraction methods can be categorized into two groups: statistical and structural (Olszewski, 2001). Regarding the statistical methods, they use quantitative characteristics of the data to extract features. Table 2.2 shows a summary of common statistical features in HAR literature. As for structural methods, however, they consider the interrelationship among data to extract features from signals. In some situations, such features play an extremely significant role in HAR. To illustrate, consider Figure 2.4 which represents the heart rate signal for a person that was walking and the same signal in reverse temporal order. Under such

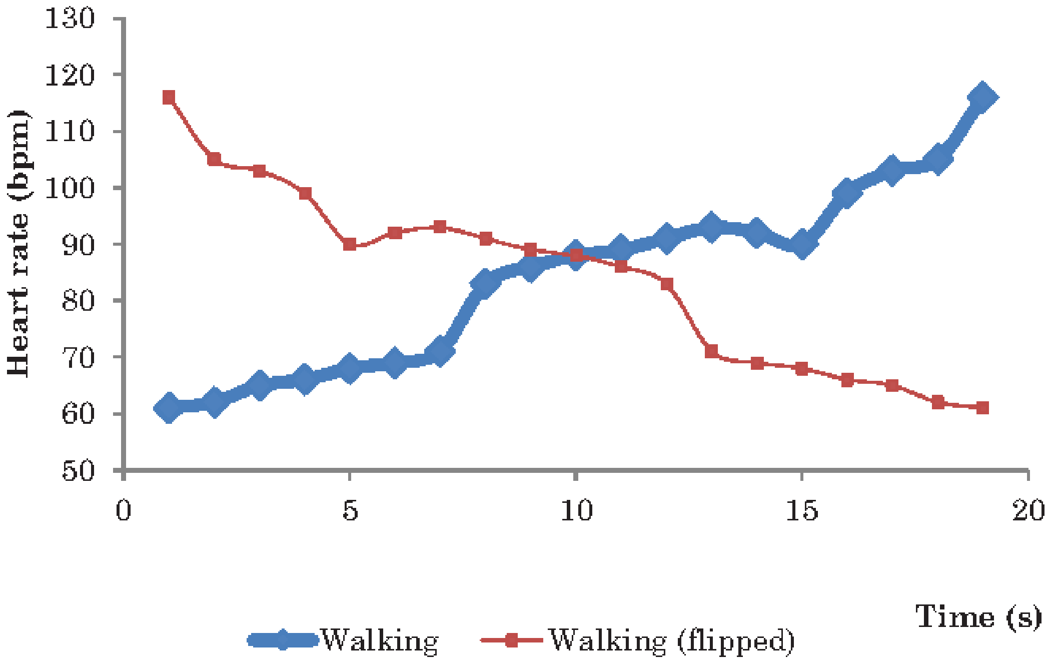


Figure 2.4: Heart rate signal for walking (bold) and flipped signal (thin) (extracted from Lara and Labrador (2012)).

circumstances, statistical features such as time domain and frequency domain features are not discriminative since for these two signals, most of them are equal and we need to use utilize features that take the structure of signals into account. According to Lara and Labrador (2012) and from a mathematical perspective, given a time series $Y(t)$, a structure detector implements a function $f(Y(t)) = \hat{Y}(t)$ such that $\hat{Y}(t)$ approximates $Y(t)$. Table 2.3 shows the common functions implemented by structure detectors. In general, choosing the feature extraction method and feature sets are dependent to the nature of signal. For instance, acceleration signals tend to fluctuate and be oscillatory and the features extraction method should be able to handle the high variability of signals.

Classification

Finally, a classifier is trained on the vector of extracted features and corresponding labels, and assigns future observations to one of the learned activities. Table 2.4 summarizes most common classifiers and some work examples of them in the HAR field.

Function	Equation	Parameters
Linear	$F(t) = mt + b$	{m, b}
Polynomial	$F(t) = a_0 + a_1 t + ... + a_{n-1} t^{n-1}$	$\{a_0, ..., a_{n-1}\}$
Exponential	$F(t) = a\|b\|^t + c$	{a, b, c}
Sinusoidal	$F(t) = a \times sin(t + b) + c$	{a, b, c}

Table 2.3: Common functions implemented by structure detectors (extracted from (Lara & Labrador, 2012)).

2.1.2 Subject dependent cross validation and subject independent cross validation

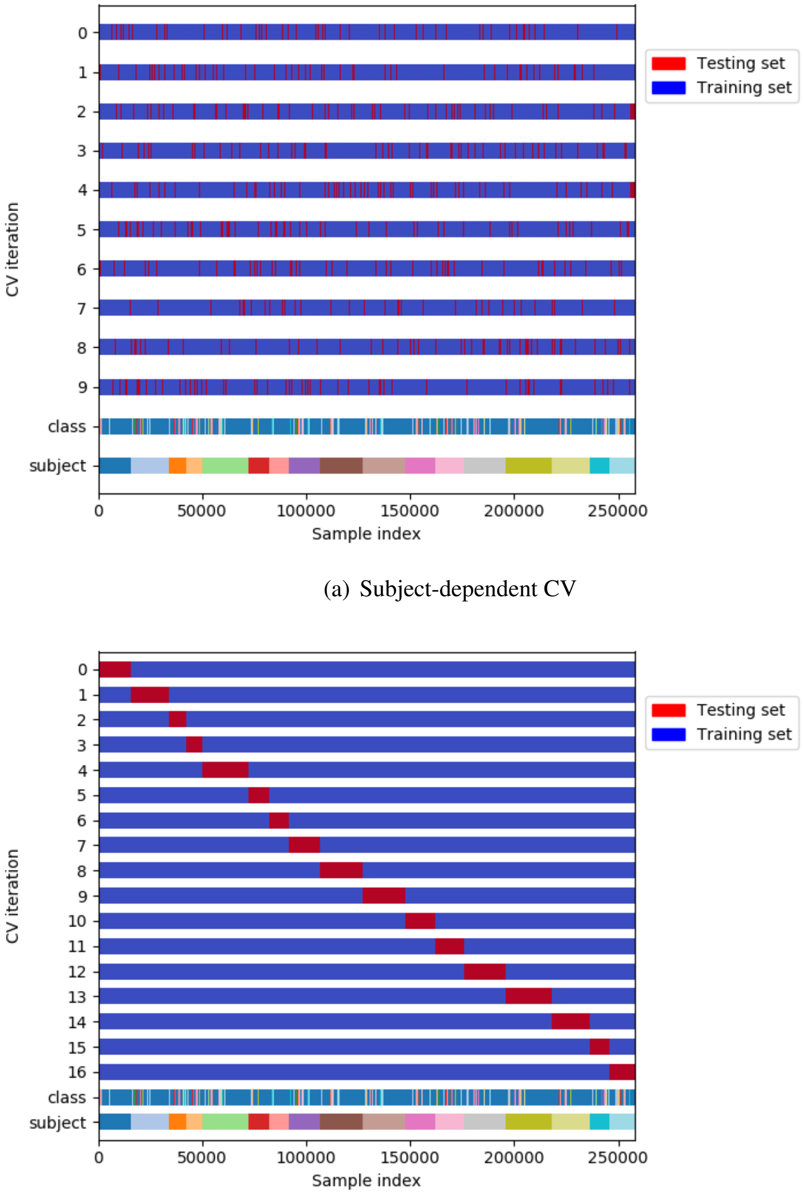
One of the most important steps in designing a system is evaluation. The evaluation in HAR has been mostly carried out through k-fold cross validation. In k-fold CV (Figure 2.5(a)), the overall data is randomly partitioned in k equal subsets. The model is then trained on $k - 1$ subsets, and the remaining one is used for testing (Trevor, Robert, & JH, 2009). In this process, the test set can be any part of the dataset meaning that training and test sets may contain the data of same subject and due to that, this method is referred to as subject-dependent CV in literature (Al Machot et al., 2019). The main assumption of this process is that *samples are Independent and Identically Distributed (i.i.d.)* (Arlot, Celisse, et al., 2010), which means that all the data points are sampled independently from the same distribution. However, samples drawn from a given subject are likely to *not* be independent, for two reasons. First, there is a strong inter-subject variability in the way activities are conducted (Bulling et al., 2014). This means that the similarity of samples drawn from the same subject is likely to be higher than that of samples drawn from different subjects. Several factors might explain such variability, including sex, gender, age or experience. Second, there is a temporal dependence between activities performed by the same subject: the similarity between samples drawn in a short time interval, for instance in the same training session in case of training activities, will most likely be higher than that of samples drawn further apart in time. This is due to

Classifier	Publications
Decision tree	Banos et al. (2014), Jatoba, Grossmann, Kunze, Ottenbacher, and Stork (2008), Maurer, Smailagic, Siewiorek, and Deisher (2006b), Bao and Intille (2004b), Ermes, Parkka, and Cluitmans (2008)
Naive Bayes	Banos et al. (2014), Tapia et al. (2007), Bao and Intille (2004a), Lara and Labrador (2012), Kern et al. (2003), Ravi et al. (2005)
K-nearest neighbors	Banos et al. (2014), Jatoba et al. (2008), Maurer et al. (2006b), Maurer et al. (2006a), Ravi et al. (2005)
Neural Networks	Sarbishei (2019), Al Machot, Elmachot, Ali, Al Machot, and Kyamakya (2019), J. Wang, Chen, Hao, Peng, and Hu (2018)
Support Vector Machines	Morris et al. (2014), Zhen-Yu He and Lian-Wen Jin (2008), J. Wang et al. (2018), Zhenyu He, Zhibin Liu, Lianwen Jin, Li-Xin Zhen, and Jian-Cheng Huang (2008), Huynh et al. (2007),Zhang and Sawchuk (2011)
Markov models	Vinh et al. (2011), Zhu and Sheng (2009)
Nearest Centroid Classifier	Huynh and Schiele (2005)

Table 2.4: Summary of common classifiers in HAR publications

factors such as fatigue and training. Thus, k-fold CV may overestimate the performance of recognizer systems in HAR. Such overestimation is even larger when k-fold CV is used with overlapping sliding windows since the overlap between adjacent windows is another source of dependency between data points. A more formal discussion about the problems of k-fold CV in HAR can be found in (Dehghani, Glatard, & Shihab, 2019).

To address these issues, the training and testing sets should be split by subject. In this method which is known as subject-independent CV (Al Machot et al., 2019; Janidarmian et al., 2017), in each iteration the model is trained on all the subjects except one, which is used for testing. In this way, the intra-subject dependencies present in subject-dependent CV are hence removed. It should be noted that, in this case, as is shown in Figure 2.5(b), the number of folds is lower or equal to the number of subjects in the dataset.



(a) Subject-dependent CV

(b) Subject independent CV

Figure 2.5: Different types of CV in HAR

2.2 Related work

The sliding window technique is widely employed in HAR, and the literature contains a plethora of examples where both, overlapping (e.g., (Bao & Intille, 2004a; Lara, Pérez, Labrador, & Posada, 2012; Morris et al., 2014; Tapia et al., 2007)) and non-overlapping time windows are used (e.g., (Banos et al., 2014; Cheng, Amft, & Lukowicz, 2010; Minnen, Westeyn, Ashbrook, Presti, & Starner, 2007; Reddy et al., 2010)).

2.2.1 Overlapping sliding windows

The work by Bao and Intille (2004a) uses overlapping sliding windows to segment signals coming from five biaxial accelerometers placed on 20 subjects (13 males and 7 females) under laboratory and semi-naturalistic conditions while performing 20 daily activities. Subsequently, each window is transformed into a set of features namely mean, energy, frequency-domain entropy, and correlation of acceleration data. The authors use k-nearest neighbor (KNN), decision tree (DT), and naive bayes (NB) classifiers, and subject-independent cross-validation (CV) and subject-dependent CV for system evaluation. They reach the overall accuracy of 84% with DT under the subject-independent CV process.

Another example is work by Tapia et al. (2007) where the authors develop a real-time recognition system to recognize physical activities and in some cases, their intensities. They segment the signal data collected from 21 subjects wearing triaxial wireless accelerometers and a wireless heart rate monitor while performing 30 physical gymnasium activities using overlapping sliding windows. Subsequently, from each window, they extract time domain and frequency domain features and using a DT classifier they recognize activities.with an accuracy of 94.6% using subject-dependent CV and 56.3% using subject-independent CV.

Lara et al. (2012) combine acceleration data with vital signs to improve the performance of HAR systems. They apply ARP on a dataset that was collected from eight subjects (7 males and 1 female) while wearing a BioHarness^{TM} BT chest sensor strap[1]. Using data from a triaxial accelerometer and vital signs, they detect five activities including running, walking, sitting, ascending, or descending.

[1] http://www.zephyr-technology.com/bioharness-bt.html

Sensor signals are partitioned into overlapping time windows with three different sizes: 5-seconds, 12-seconds, and 20-seconds sliding at 50% of their sizes and 90 features were extracted from each time window. NB, DT, Bayesian Network, Multilayer Perceptron, Additive Logistic Regression and classifier ensembles are used to recognize activities. They achieve up to 95.7% overall accuracy, which was evaluated through subject-dependent CV. Their results also indicate that vital signs are useful to discriminate between certain activities like running and sitting compared to the cases that utilize acceleration data only.

Recofit (Morris et al., 2014) is a well-known reference on HAR, which applies ARP on a dataset of accelerometer and gyroscope data collected from 114 participants over 146 sessions. The authors address three major challenges namely (1) segmenting exercise from intermittent non-exercise periods, (2) recognizing which exercise is being performed, and (3) counting repetitions. Data points are windowed into 5-second overlapping windows sliding at 200 ms and subsequently, each window is transformed into 224 features. Linear support vector machines (SVM) are used in the classification stage and evaluated by subject-independent CV. Spectacular performance is achieved, with precision and recall greater than 95% to identify exercise periods, recognition of up to 99% for circuits of 4 exercises, and counting accurate to ±1 repetition, 93% of the time.

2.2.2 Non-overlapping sliding windows

Regarding non-overlapping windowing, Minnen et al. (2007) describes an activity recognition component to recognize soldier activities as a part of the Soldier Assist System (SAS). They apply ARP on the signal of a six three-axis bluetooth accelerometers positioned on the right thigh sidearm holster to recognize 14 soldier activities. The sensor signals are partitioned by 3-second non-overlapping sliding windows and then each window is transformed into 378 features. A boosting ensemble classifier is used to select the most important features and also recognize the activities. Their recognition system achieves 78.7% for continuous event recognition (considering null activity) and 70.3% frame level accuracy. These values increase to 90.3% and 90.3%, respectively when considering only the modeled activities. In their study, they use subject-independent CV to evaluate their system.

Another example is the work by Reddy et al. (2010), where the authors create a transportation mode recognition system using a mobile phone to identify whether an individual is stationary, walking, running, biking, or in motorized transport. The dataset used in their study contains accelerometer measurements, GPS, WiFi, and GSM signals of sixteen individuals (eight male and eight female) while six phones attached to their bodies simultaneously and were in one of the five transportation modes for fifteen minutes. The signals are windowed into 1-second non-overlapping sliding windows and each window is transformed into a set of features such as magnitude of the force vector, mean, variance, energy, etc. They use several classifiers namely DT, NB, KNN, SVM, Hidden Markov Model and a two-stage classifier involving DT combined with discrete Hidden Markov Model. They achieve an accuracy level of 93.6% with the two-stage classifier, which was evaluated by subject-dependent CV.

Cheng et al. (2010) implement an on-body capacitive sensing approach to recognize activities such as chewing, swallowing, speaking, sighing (taking a deep breath), as well as different head motions and positions. They use a dataset that contains the 4.3 hours-electrode collar data which was collected from three subjects (one female, two males; aged between 25 and 45 years) while performing a set of head movements, swallow water from a cup, chew and swallow bread pieces and speak. Each signal is partitioned into 1.5-second non-overlapping sliding windows and each window then is transformed into time domain features such as signal mean, variance, maximum, etc. They use a linear discriminant classifier to identify various activities and evaluate the system through subject-dependent CV and report the accuracy rate for the combination of activities.

Another example is the work by Banos et al. (2014), where the authors present an extensive study to distinguish the windowing procedure, its impacts on the recognition system. They apply ARP on the accelerometer data of a benchmark dataset collected from 17 subjects of different profiles performing 33 fitness activities in an out-of-lab environment. Sensor signals are windowed into non-overlapping windows with a substantial set of window sizes ranging from 0.25 to 7 seconds in steps of 0.25-seconds. Each window is then transformed into three different feature sets (FS) namely FS1 (mean only), FS2 (mean and standard deviation) and FS3 (mean, standard deviation, maximum, minimum and mean crossing rate). They use DT, KNN (K=3), NB and Nearest Centroid Classifier (NCC) as the classifiers and subject-dependent CV for system evaluation. From this study, they prove that the interval 1–2 second is the best trade-off between recognition speed and performance. Besides, they provide a set of guidelines for system definition and configuration based on the particular application requirements and target activities.

2.3 Summary

In this chapter, all stages of ARP and the commonly used techniques in each stage are reviewed in details. Considering sliding windows as the most widely used technique in the segmentation stage of ARP, we review different types of this technique in previous works in HAR literature. In the next chapter, we present our experiment setting including datasets, features sets, window sizes, etc.

Chapter 3

Methodology

3.1 Datasets

In this study, we use two public datasets of human activity, to evaluate the impact of overlapping windows in a wide range of subjects, activities, and conditions.

3.1.1 Dataset 1

As the first dataset, we use the dataset described in (Baños et al., 2012), one of the most complete public datasets for HAR in terms of the number of activities and subjects. The dataset consists of data collected from 17 subjects of diverse profiles while wearing 9 Xsens[1] inertial measurement units (IMU) on different parts of their body. Subjects performed 33 fitness activities (Table 3.1) ranging from warm up to fitness exercises in an out-of-lab environment. Each sensor provides tri-directional acceleration, gyroscope, and magnetic field measurements, as well as, orientation estimates in quaternion format (4D). Similar to prior work in Banos et al. (2014), acceleration data was used in our study. The dataset also provides data for three sensor displacement scenarios namely “default”, “self-placement” and “mutual-displacement” to compare the sensor anomalies, but as in Banos et al. (2014), only the data from the default scenario is used in our study.

[1] https://www.xsens.com

Activity	Label	Activity	Label
No activity	0	Upper trunk and lower body opposite twist (20x)	18
Walking (1 min)	1	Arms lateral elevation (20x)	19
Jogging (1 min)	2	Arms frontal elevation (20x)	20
Running (1 min)	3	Frontal hand claps (20x)	21
Jump up (20x)	4	Arms frontal crossing (20x)	22
Jump front & back (20x)	5	Shoulders high amplitude rotation (20x)	23
Jump sideways (20x)	6	Shoulders low amplitude rotation (20x)	24
Jump leg/arms open/closed (20x)	7	Arms inner rotation (20x)	25
Jump rope (20x)	8	Knees (alternatively) to the breast (20x)	26
Trunk twist (arms outstretched) (20x)	9	Heels (alternatively) to the backside (20x)	27
Trunk twist (elbows bended) (20x)	10	Knees bending (crouching) (20x)	28
Waist bends forward (20x)	11	Knees (alternatively) bend forward (20x)	29
Waist rotation (20x)	12	Rotation on the knees (20x)	30
Waist bends (reach foot with opposite hand) (20x)	13	Rowing (1 min)	31
Reach heels backwards (20x)	14	Elliptic bike (1 min)	32
Lateral bend (10x to the left + 10x to the right)	15	Cycling (1 min)	33
Lateral bend arm up (10x to the left + 10x to the right)	16	-	-
Repetitive forward stretching (20x)	17	-	-

Table 3.1: Activity set in dataset 1.

3.1.2 Dataset 2

The second dataset used is also one of the most complete and big public datasets for HAR in terms of the number of activities and subjects (Morris et al., 2014). The dataset contains data for 74 activities (Table 3.2) from 94 subjects (28 female), ages 18-58 ($\mu = 34.2$). The data was collected in a large lab space from an armband worn on the right forearm, containing a SparkFun “Razor IMU” inertial sensor[2]. This IMU includes a 3-axis accelerometer and a 3-axis gyroscope, which transmit sensor values to a PC at 50Hz. As can be seen in Table 3.2, there are several activities in the dataset such as ”Arm band adjustment” and ”Device on Table” which we consider as noise in this study. Besides, since during the data collection, the subjects wore a single joint sensor on their right forearm, the activities of the opposite hand (left hand) can not be captured properly with the sensors. Thus, we have done a relabeling process to clarify the dataset. In this process, (1) we label all the irrelevant and opposite hand activities as a ”Noise” class (2) all the activities that refer to multiple labels are grouped together as one exercise. Table 3.2 shows all the activities and their labels after the relabeling process.

3.2 Experimental Setup

Similar to prior work by Banos et al. (2014), we did not apply any pre-processing to the dataset. We used both overlapping and non-overlapping windows. Overlapping windows were sliding at 200 ms, with window sizes ranging from 0.25 s to 7 s in steps of 0.25 s. For instance, a 5-second window shared 4.8 s of data with the previous one. Given the constant value of the sliding duration (200 ms), using a set of different window sizes is equivalent to exploring the impact of various overlapping sizes on the performance of our HAR systems. For non-overlapping windows, we used the same settings as in Banos et al. (2014): disjoint windows with sizes ranging from 0.25 s to 7 s in steps of 0.25 s. We used the same feature sets as in Banos et al. (2014), namely FS1 (mean only), FS2 (mean and standard deviation) and FS3 (mean, standard deviation, maximum, minimum and mean crossing rate). Finally, for the classification part, we used the following classifiers: DT, KNN (K=3), NB and Nearest Centroid Classifier (NCC). We used these classifiers as implemented

[2] https://www.sparkfun.com

Activity	Label	Activity	Label	Activity	Label
Arm band adjustment	1(Noise)	Lawnmower (both)	20	Squat (arms in front)	33
Arm straight up	1(Noise)	Lawnmower (left)	1(Noise)	Squat (hands behind head)	33
Band Pull-Down row	2	Lawnmower (right)	21	Squat (kettlebell)	33
Bicep Curl	3	Lunge (both legs)	22	Squat Jump	33
Bicep Curl (band)	3	Ball Slam	23	Squat Rack Shoulder Press	33
Box Jump	4	No Exercise	1(Noise)	Static Stretching	1(Noise)
Burpee	5	Note	1(Noise)	Stretching	1(Noise)
Butterfly sit-up	6	Triceps Extension(standing)	24	Tap IMU	1(Noise)
Chest Press	7	Triceps Extension (both)	24	Tap left IMU	1(Noise)
Crunch	8	Plank	25	Tap right IMU	1(Noise)
Device on Table	1(Noise)	Power Boat pose	26	Triceps Kickback(bench –both)	34
Dip	9	Pushups (foot variation)	27	Triceps Kickback (bench –left)	1(Noise)
Dumbbell Deadlift Row	10	Pushups	27	Triceps Kickback (bench –right)	34
Dumbbell Row (both)	11	Stretching	1(Noise)	Triceps Extension (lying –both)	35
Dumbbell Row (left)	1(Noise)	Rest	1(Noise)	Triceps Extension (lying –left)	1(Noise)
Dumbbell Row (right)	12	Rowing Machine	28	Triceps Extension (lying –right)	35
Dumbbell Squat (hands at side)	13	Running	29	Two-arm Dumbbell Curl (both)	36
Dynamic Stretch	1(Noise)	Russian Twist	30	Non-listed	1(Noise)
Elliptical Machine	14	Seated Back Fly	31	V-up	37
Punches	15	Shoulder Press	32	Walk	38
Invalid	1(Noise)	Side Plank (left)	25	Walking lunge	39
Jump Rope	16	Side Plank (right)	25	Wall Ball	40
Jumping Jacks	17	Sit-up (hand behind head)	6	Wall Squat	41
Kettlebell Swing	18	Sit-up	6	Dumbbell Curl (alternating)	36
Lateral Raise	19	Squat	33		

Table 3.2: Activity set in dataset 2.

in scikit-learn 0.20 (Pedregosa et al., 2011) with default hyperparameters. We have also investigated non-default hyperparameters in another experiment. Furthermore, to provide a comparison with the framework in Sarbishei (2019), we have utilized time-domain histogram features and the AdamOptimizer algorithm in Tensorflow 1.12.0 to train a neural network classifier with Sigmoid and Softmax trigger functions in the hidden and output layers, respectively. Such experiments will be explained in details in Chapter 4.

To evaluate model performance, we used both subject-dependent CV and subject-independent CV. We use the F1-score as a performance measure due to its robustness in class imbalance. F1-score which reaches its best at 1 and worse at 0, is computed as follows:

$$F1 = \frac{2 \times (precision \times recall)}{(precision + recall)}$$

All the source code for the conducted experiments are available in our GitHub repository [3]. The repository contains the scripts to segment the datasets for different window sizes, feature sets and sliding window techniques. There is also a script for training and testing all mentioned classifiers on windowed datasets. Finally, it also contains code to reproduce all presented figures in this paper.

3.3 Classifiers

In this section, the utilized classifiers in this study are explained.

3.3.1 Decision Tree

Decision Tree is a non-parametric supervised learning method used for classification and regression (Quinlan, 1986). A decision tree is a flowchart-like structure which is constituted of several nodes and branches. Nodes can be categorized into two types namely internal nodes (decision Node) which show a test on a feature of dataset and leaves (terminal Node) which represent a class label of the dataset. Regarding the branches, they represent the output of the test on internal nodes. A tree is constructed by splitting the training dataset into subsets based on a set of tests on the values of a certain feature. This process follows a recursive manner meaning that for each derived subset it is repeated. The recursion is finished when the classes of all samples in the subset at a node are

[3] http://www.github.com/big-data-lab-team/paper-generalizability-window-size

the same or when partitioning the data does not improve the prediction. The primary challenge in the growing the decision tree (learning phase) is to select the best attribute as the root node and also each level to split the dataset. To do so, the impurity of each node should be calculated. Table 3.3 indicates the most widely used criterion for measuring the node impurity. Using such techniques, at each internal node, the entropy of all attributes in the dataset is calculated and one with the lowest entropy is selected to split the dataset. In scoring time for a given data point, we start from the root of the tree and apply its test on the corresponding attribute of data point. Based on the result, we follow the corresponding branch and jump to the next level. We continue until we reach a leaf node that contains predicted class value.

Criterion	Formula	Description
Gini Index	$\sum_i p(i)(1 - p(i))$	p(i) is the probability that an arbitrary sample being classified to class i.
Entropy	$-\sum_i p(i)\ log\ p(i)$	p(i) is the relative frequency of class i
Towing rule	$\frac{p_L p_R}{4}(\sum_i \lvert(p(i \mid t_L) - (p(i \mid t_R))\rvert)^2$	where L and R refer to the left and right sides of a given split respectively, and $p(i \mid t)$ is the relative frequency of class i at node t

Table 3.3: The most common utilized techniques for splitting the dataset in DT (Zambon et al., 2006)

3.3.2 K-Nearest Neighbors

KNN is an instance-based supervised learning method used for classification and regression (Cover, Hart, et al., 1967). This method is also categorized as non-generalizing learning algorithms since it does not learn a general internal model from the training dataset but simply stores all of the data points. KNN is based on a neighborhood majority voting scheme and assigns the new instance to the most common class amongst its K nearest neighbors. Table 3.4 describes several metrics to measure the similarity of two vectors which can be used to find the k nearest neighbors of a given instance. Regarding the number of neighbors (K), according to to ("Miscellaneous Clustering Methods", 2011), increasing it may (1) improve the performance of the classifier (2) reduce the effect of noise on the classification (3) makes the decision boundary less distinct. It should be noted that choosing a too big value for k may result in performance drop since it can destroy the locality and as a result, KNN uses the not neighbors to classify the coming data point.

Distance Metric	Formula	Distance Metric	Formula
Euclidean	$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$	Standardized Euclidean	$\sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{{x_i}^2}}$
City Block	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$	Minkowski	$\sqrt[p]{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p}$
Chebychev	$\max_i \{\lvert x_i - y_i \rvert\}$	Cosine	$\frac{x.y}{\lvert\lvert x \rvert\rvert \lvert\lvert y \rvert\rvert}$

Table 3.4: Common used distance metrics in KNN - x and y are two vectors with equal length

3.3.3 Naïve Bayes

NB (Theodoridis & Koutroumbas, 1998) is a supervised learning algorithm based on the Bayes theorem with the naive assumption of feature independence. The following equations show the NB formula:

$$P(y|x_1, x_2, ..., x_n) = \frac{p(y) \prod_{i=1}^{n} p(x_i|y)}{p(x_1, x_2, ..., x_n)} \tag{1}$$

Where y represents classes and x_1 to x_n are feature vector. Since $p(x_1, x_2, ..., x_n)$ is constant given the input, we can use the following classification rule:

$$\hat{y} = argmax_y p(y) \prod_{i=1}^{n} p(x_i|y) \tag{2}$$

In general, there are several types of NB classifiers such as Gaussian NB, Multinomial NB and Complement NB which mostly differ by the assumptions they make about the distribution of $p(x_i|y)$.

3.3.4 Nearest Centroid Classifier

NCC (Tibshirani, Hastie, Narasimhan, Chu, et al., 2003) is a classification model which widely used in text classification (Tan, 2008), bioinformatics (Levner, 2005) and medical domain (Sharma

& Paliwal, 2010). This classifier calculates the centroid (mean) of each class in training set and assigns to the observation the class whose mean is closet to the observation. The following equations show the training and predicting phases of NCC:

Training: given labeled training samples $\{(\vec{x}, y_1), (\vec{x}, y_2), ..., (\vec{x}, y_n)\}$ where $\vec{x}$ is feature vector and y_i is the class label ∈ Y, for each class the centroid is calculated as follows:

$$\vec{u_l} = \frac{1}{c_l} \sum_{i \in c_l} \vec{x_i} \tag{3}$$

where C_l is the set of indices of samples belonging to class l ∈ $\mathbf{Y}$.

Prediction: The assigned class ($\hat{y}$) to a given observation ,$\vec{x_i}$ calculated as follow:

$$\hat{y} = argmin_l ||\vec{u_l} - \vec{x}|| \tag{4}$$

3.3.5 Artificial Neural Network

An Artificial Neural Network (ANN) is a systematic procedure of data processing inspired by the nervous system function in animals. It tries to reproduce the brain logical operation using a collection of neuron-like entities to perform processing of input data (Cortina-Puig et al., 2010). The basic processing unit of an ANN is called perceptron which is an approximation to the biological neuron. It is a decision-making unit with several input connections and a single output as shown in Figure 3.1. The input neuron p_i is weighted with a connection weight w_i w_i. The perceptron sums the dot product of weights and inputs vectors and adds a bias b. The obtained total value (n) will be transformed by a function (f) to produce the output value. This process is summarized as:

$$a = f(b + \sum_{j=1}^{i} w_j p_j) \tag{5}$$

The function f which is called activation function or transfer function is used to introduce non-linear properties into the network. Tables 3.5 shows the most widely used activation functions.

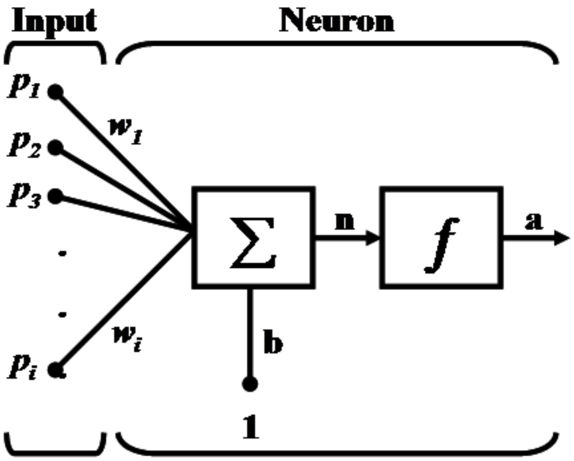


Figure 3.1: Schematic representation of the perceptron (extracted from Cortina-Puig et al. (2010)).

Activation function	Formula
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Hyperbolic Tangent	$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$
Rectified Linear units	$f(x) = max(x, 0)$

Table 3.5: Commonly used activation functions in ANN

The problem of a perceptron is that it can not learn nonlinearly separable datasets. To solve this problem, we can add several perceptrons to form a layer and also introducing further layers. In this way, we have a multilayer feedforward neural network. In general, there are three types of layers in ANN namely input layer, hidden layer, and output layer. The goal of the input layer is to distribute incoming signals to the next layer and by definition, its units are not perceptron. However, the units in all layers after the input layer and output layer are perceptron since they communicate with the environment by sending or receiving messages to units in the layers to which they are connected. The output layer provides a link between the network and environment by submitting the processed information. In a classification problem, the number of units in the output layer is equal to the number of class labels Sharma, Lee, and Chung (2008). In training the network, its parameters (w_i) are adjusted so that the difference between the output units and the target values is minimized. There are several ways to train an ANN such as Resilient Backpropagation Riedmiller and Braun (1993), Scaled Conjugate Gradient Møller (1993) and Levenberg–Marquardt backpropagation (Hagan & Menhaj, 1994).

3.4 Summary

In this chapter, the used datasets in this thesis are discussed. Subsequently, we explain the experiment setup including feature sets, window sizes, classifiers, etc. As discussed in Chapter 2, there are numerous classifiers that have been used in HAR field. In this chapter, we tried to cover those that we utilized in this work. In the next chapter, we explain the conduced experiments and present their results.

Chapter 4

Results

In this section, the impact of the overlapping and non-overlapping sliding windows in HAR systems with Subject-dependent CV and Subject-independent CV for both datasets is evaluated. The experiments are categorized into global evaluation and activity-specific analysis.

For each experiment, we report distributions of average F1-score values obtained across all validation folds. Each distribution contains 28 measures obtained for the different window sizes mentioned in Section 3.2. We selected this representation over summary statistics or confusion matrix as it provides a comprehensive and synthetic overview of classification performance over a range of window sizes, classifiers, and feature sets. In all the figures of this section, “O” and “NO” stand for overlapping and non-overlapping windowing respectively. Non-overlapping windowing is represented in green, and overlapping windowing is in red.

4.1 Global evaluation

In this set of experiments, we analyze the general impact of overlapping and non-overlapping windowing in HAR systems trough the average performance of models for different activities and window sizes.

Experiment 1: Subject-dependent CV

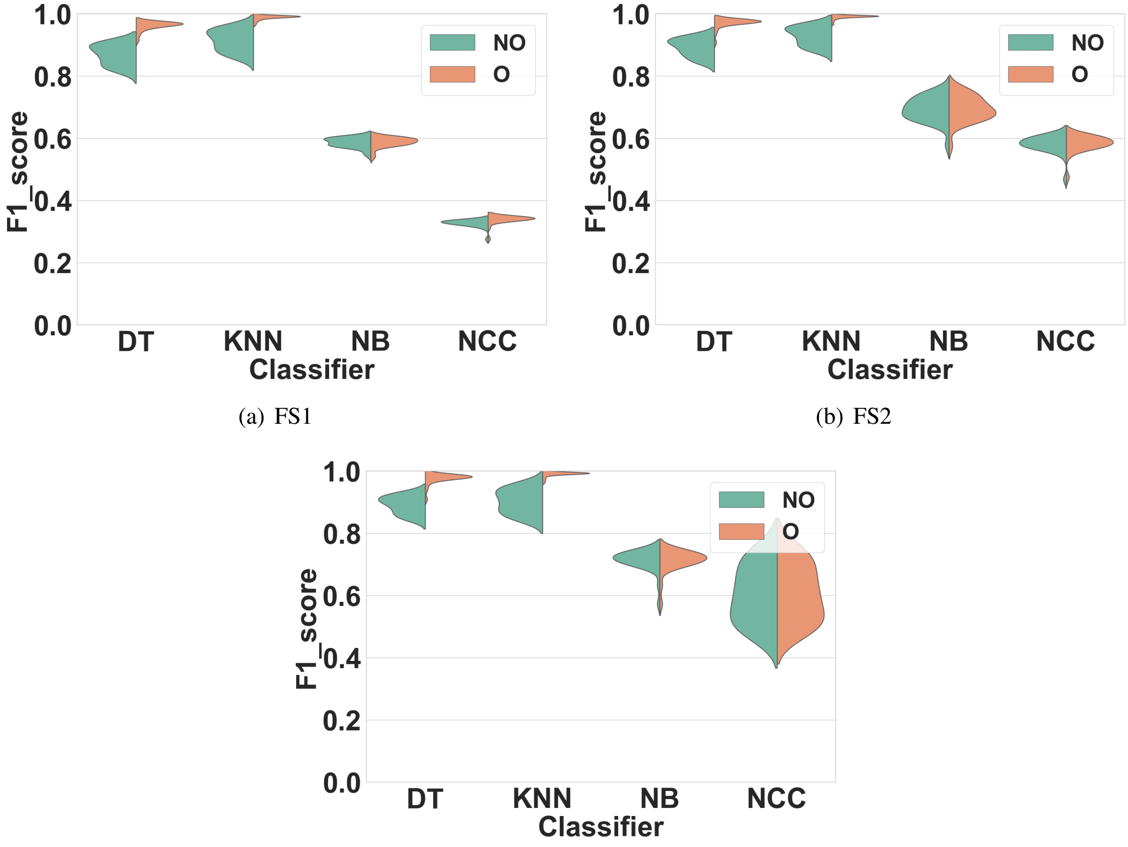
In this experiment, we apply non-overlapping and overlapping windowing with subject-dependent CV and use it as a baseline for further evaluations.

We applied the ARP system as explained in Section 3.2, on the datasets described in Section 3.1. For each window size, we partitioned the dataset in non-overlapping and overlapping windows separately and extracted feature sets FS1, FS2, and FS3 in each window. We trained the classifiers on the resulting feature vectors, and measured their average F1-score over 10-fold CV.

Dataset 1. Figure 4.1 shows the distribution of the F1-scores of the classifiers for different window sizes in overlapping and non-overlapping windows. The classifiers can be categorized into two groups: (1) KNN and DT, that have very different performance distributions for overlapping and non-overlapping windowing, and (2) NB and NCC, that show almost similar distributions for both techniques. Our findings show that, in general, using overlapping windowing improves the performance of all classifiers in all feature sets. Regarding the first group (KNN and DT), quantitatively, using the overlapping windowing technique improves the F1-score of the KNN and DT by about 10%, 8% and 8% on average in FS1, FS2, and FS3 respectively. However, the improvement for the second group is about 1%, on average, for all features sets, which is insignificant.

Dataset 2. The distribution of F1-scores for different window sizes and classifiers for overlapping and non-overlapping windowing is shown in Figure 4.2. Generally, the trends for Dataset 2 and Dataset 1 are similar. Overlapping windowing increases the F1-score and we observe the same two performance groups as before. The F1-score distributions of DT and KNN for overlapping and non-overlapping windowing are very different. Quantitatively, using overlapping sliding windows increases the performance of KNN and DT by about 9%, 12% and 13% on average in FS1, FS2, and FS3 respectively. Regarding the NB and NCC, however, the increase is minor to negligible for all feature sets.

These results show that using the overlapping windowing technique rather than the non-overlapping one in subject-dependent CV improves the performance of classifiers. This agreement between our results and the general argument of the effectiveness of overlapping windowing in HAR systems (Janidarmian et al., 2014, 2017) reinforces our confidence in the correctness of our analysis method and its applicability to our next experiments.



(a) FS1

(b) FS2

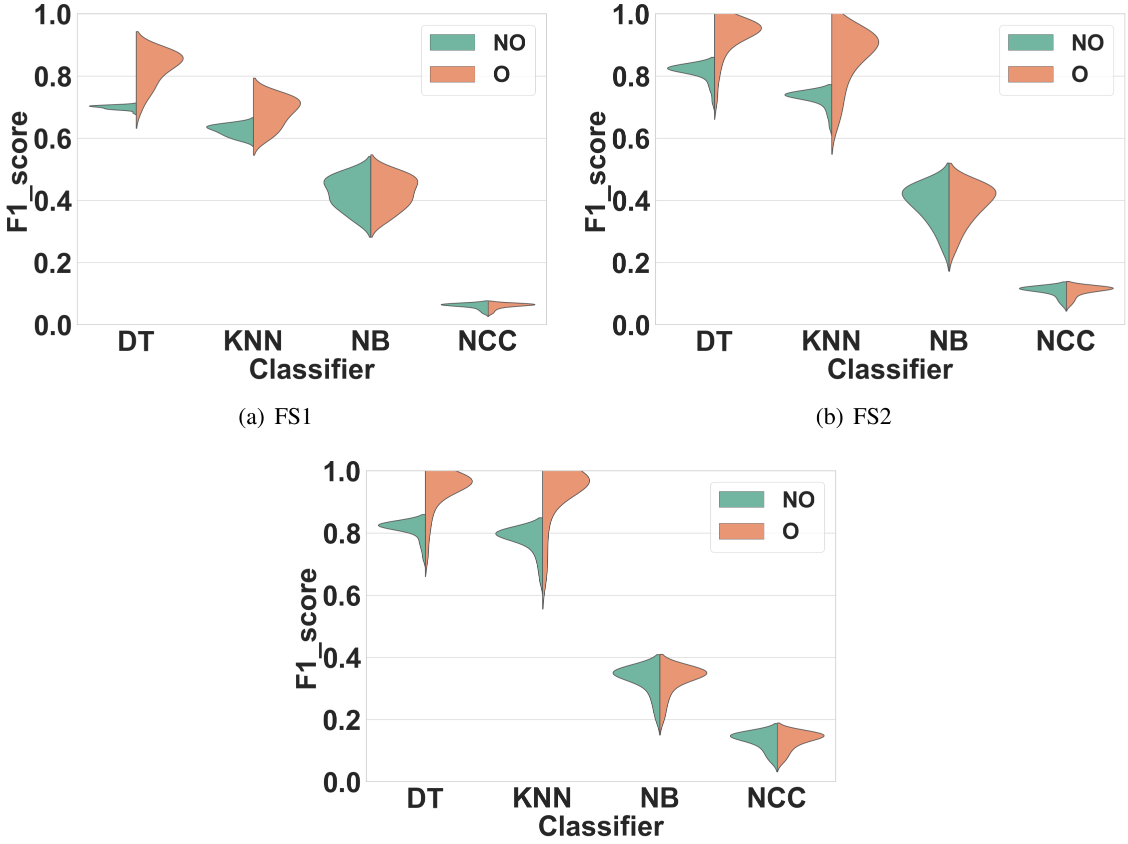
(c) FS3

Figure 4.1: Experiment 1 – Subject-dependent CV – Dataset 1.

Experiment 2: Subject-independent CV

As explained in Section 2.1.2, subject-independent CV should be used to evaluate the performance of HAR systems. Thus, in this experiment, we compare the overlapping and non-overlapping windowing techniques when using subject-independent CV. The only difference between this experiment and Experiment 1 is the use of subject-independent CV rather than subject-dependent CV.

Dataset 1. Figure 4.3 shows our results. Similar to Experiment 1 for this dataset, we observed the same two performance groups among classifiers. Regarding the first group (KNN and DT), however, overlapping windows do not lead to any improvement of the F1-score compared to non-overlapping windows. Overlapping windows even slightly decrease the F1-scores of DT and KNN in all feature



(a) FS1

(b) FS2

(c) FS3

Figure 4.2: Experiment 1 – Subject-dependent CV – Dataset 2.

sets, on average by 2% (FS1), 4% (FS2) and 1% (FS3). To further illustrate this result, Table 4.1 shows the F1-score of DT and KNN for several window sizes in overlapping and nonoverlapping windowing techniques for FS3: the performance of overlapping and non-overlapping windows is very similar. For NB and NCC the F1-scores obtained with both techniques are similar for all feature sets.

Dataset 2. Figure 4.4 shows the results of Experiment 2 for Dataset 2. Once again, the F1-scores obtained for overlapping and nonoverlapping windows are very similar. This time, with DT and KNN, overlapping windowing improves the F1-scores slightly compared to non-overlapping windowing, on average by 2% (FS1), 1% (FS2) and 1% (FS3). The comparison between the performance of KNN and DT for several window sizes is shown in Table 4.1. For NB and NCC the F1-scores obtained with both techniques are similar for all feature sets.

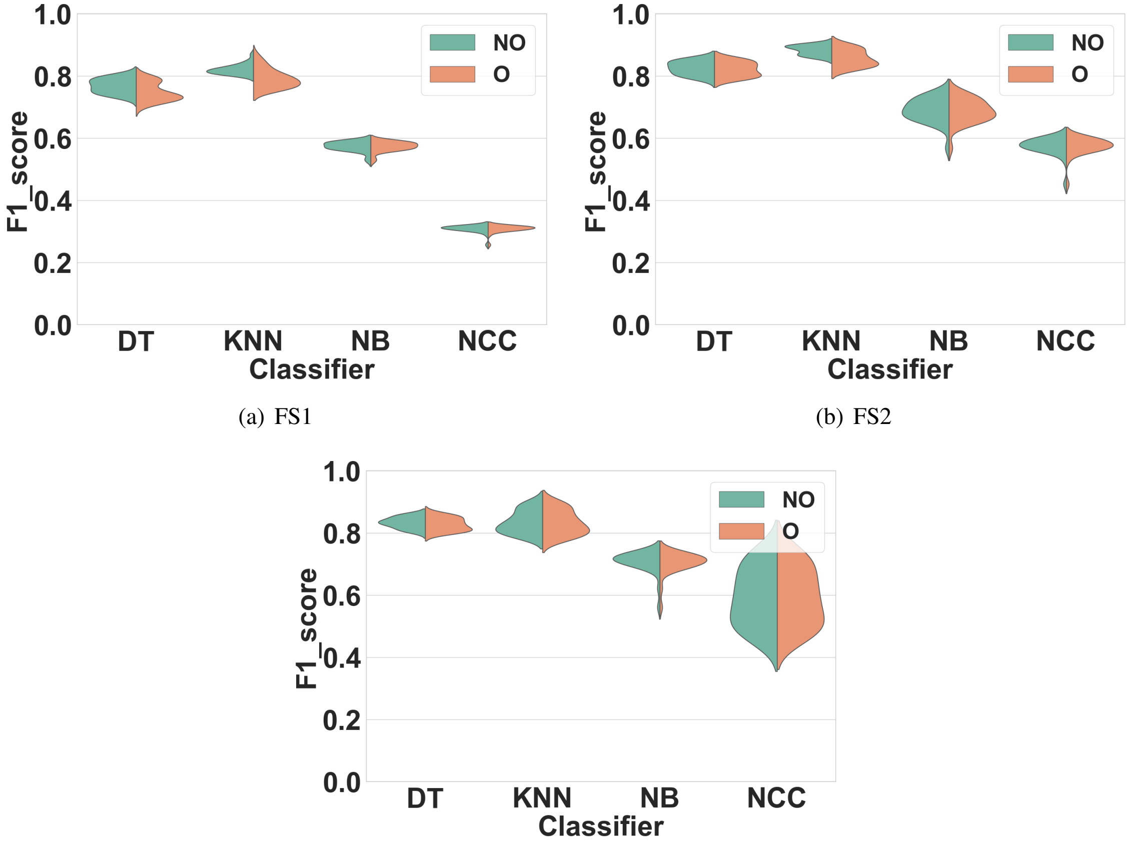
Dataset	window size (sec)	Classifier	F1-score - O (%)	F1-score - NO (%)
1	1	DT	86.42	86.0
		KNN	89.04	88.78
	4	DT	81.97	83.38
		KNN	82.05	82.18
	6	DT	80.1	80.83
		KNN	79.08	79.30
	7	DT	80.61	80.62
		KNN	77.74	78.68
2	1	DT	69.26	67.38
		KNN	63.61	63.94
	4	DT	74.37	73.06
		KNN	65.52	67.01
	6	DT	75.28	73.12
		KNN	66.0	67.8
	7	DT	75.46	73.06
		KNN	66.0	67.93

Table 4.1: F1-scores of DT and KNN for several window sizes in overlapping (O) and nonoverlapping (NO) windowing – Subject-independent CV – FS3.

In general, compared to Experiment 1 for both datasets, the performance of all classifiers in all feature sets decreased which may be due to the overestimation of subject-dependent CV. In other words, subject-independent CV removes the performance improvement resulting from using overlapping windows in Experiment 1 which resulted from the overestimation of subject-dependent CV. This experiment shows that the performance advantage of overlapping windows in the literature is due to the use of subject-dependent CV and in case of using subject-independent CV, this method does not offer any benefit to the performance of HAR systems. Hence, we can reach to the same recognition performance by using non-overlapping windows. Considering the resource-intensity of overlapping windowing compared to non-overlapping one, this is an important conclusion since through that we can save a lot of resources (energy, time, etc) which is a desirable feature in HAR (Lara & Labrador, 2012).

Experiment 3: Subject-independent CV and new hyperparameters

One may claim that the results of Experiment 2 are due to the specific set of hyperparameters used in the classifiers. Thus, to investigate that, we reproduced Experiment 2 with a new set of hyperparameters for the KNN and DT classifiers.



(a) FS1

(b) FS2

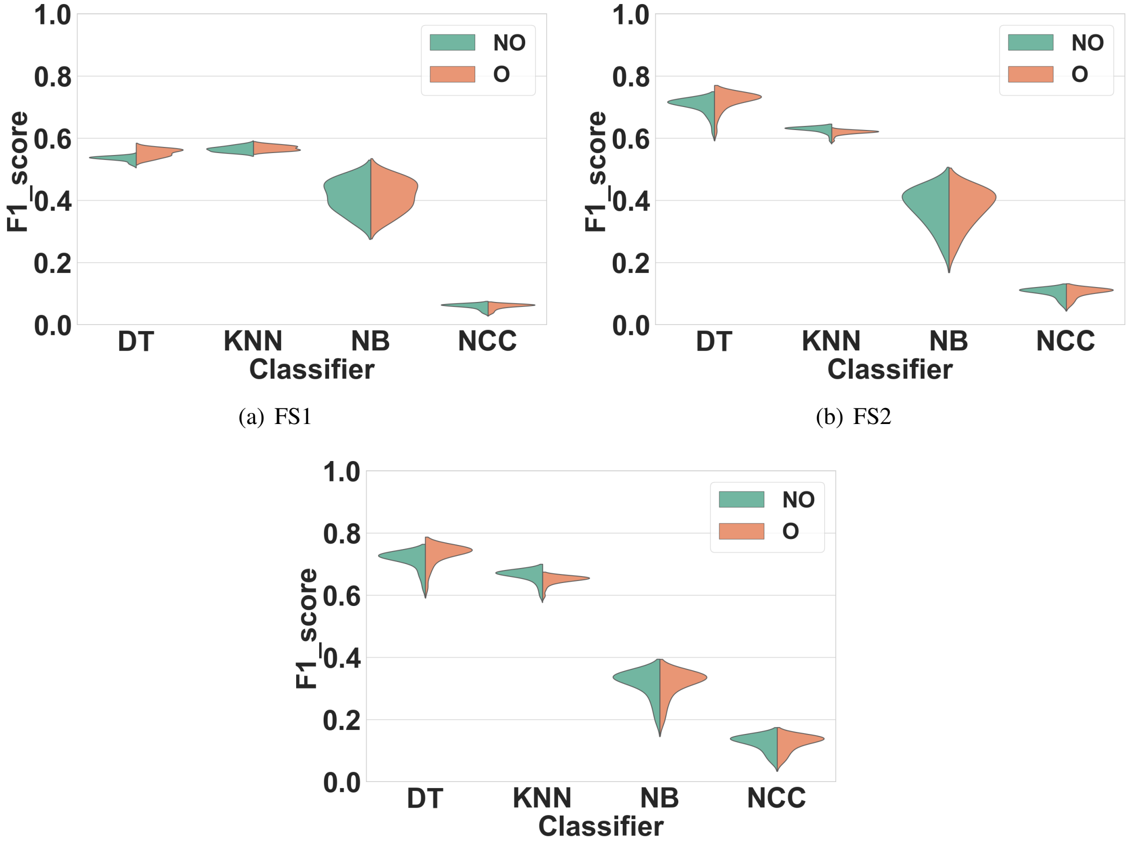
(c) FS3

Figure 4.3: Experiment 2 – Subject-independent CV – Dataset 1.

We selected new hyperparameter values such that (1) overfitting or underfitting does not occur, and (2) the new values are as different as possible from those in the previous experiments. Table 4.2 compares the selected values for hyperparameters of DT and KNN in Experiment 2 and Experiment 3.

Dataset 1. Results are shown in Figure 4.5. As in the previous experiments, the F1-scores obtained with overlapping and with non-overlapping windows are comparable. For DT, using overlapping windows decreases the performance in all feature sets, by about 1%. As for KNN, using overlapping windowing reduces the F1-score by about 4% in FS1 and FS2, and increases it by 1% in FS3.

Dataset 2. Figure 4.6 shows our results for this dataset. The trend is similar to the previous experiments, i.e., F1-scores obtained with overlapping and with non-overlapping windowing are very similar. Qualitatively, such differences for both classifiers remain lower than 1% in all feature sets.



(a) FS1

(b) FS2

(c) FS3

Figure 4.4: Experiment 2 – Subject-independent CV – Dataset 2.

In conclusion, overlapping windowing does not provide any performance improvement compared to non-overlapping windowing with our new set of hyperparameters, which confirms the findings of Experiment 2.

4.2 Activity-specific evaluation

The global evaluation presented previously is useful to have a general view of the effect of windowing techniques in HAR systems. However, it is also interesting to particularize this study to each specific activity. Thus, in this section, we analyze the impact of overlapping and non-overlapping windowing for each activity.

Classifier	Hyperparameter	Experiment 2	Experiment 3
KNN	K (n_neighbors)	3	6
DT	Criterion	Gini	Entropy
	max_depth	None	20
	max_features	1	0.7

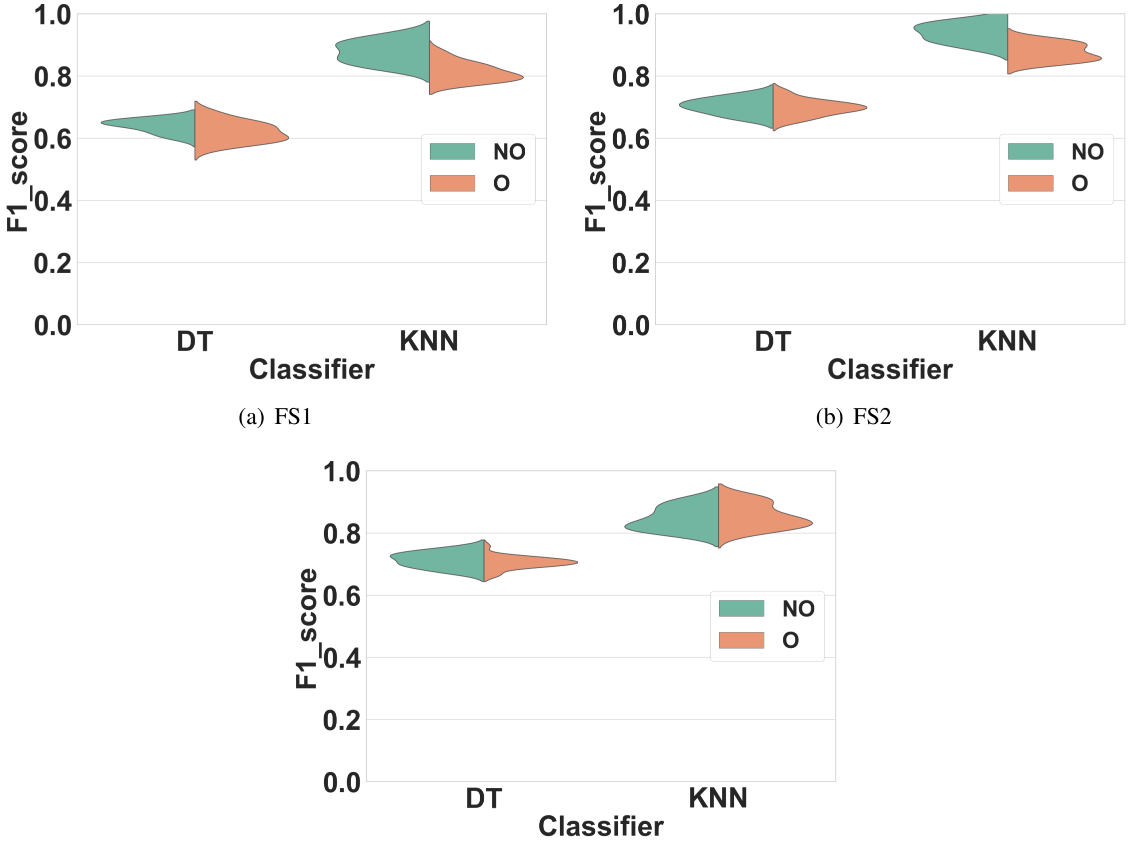
Table 4.2: The hyperparameters values for KNN and DT in Experiment 2 and Experiment 3. When max_depth is set to None, the decision tree is expanded until all leaves are pure (Pedregosa et al., 2011).

Experiment 4: Subject-independent CV per activity

As shown by Experiment 2, overlapping and non-overlapping windowing techniques only lead to minor performance differences when evaluated with subject-independent CV. In this experiment, we investigate how this result particularize to specific activities. The presented data is the same as in Experiment 2, but the classification performance is now detailed for each activity. For brevity, we only focus on feature set FS3. In all figures, activities are shown with the labels reported in Tables 3.1 (Dataset 1) and Tables 3.2 (Dataset 2).

Dataset 1. In Figure 4.7, the activity-specific F1-scores distributions achieved for all classifiers, window sizes and windowing approaches are presented. As expected from the results shown in 4.3(c), the differences between overlapping and non-overlapping windowing for all activities are slow. In general, the performances of the majority of the activities drop (by 5% on average) when overlapping windowing is used instead of non-overlapping windowing. However, using DT (Figure 4.7(a)), some activities such as Heels (27), Rotation on knees (30), Trunk Twist (10), Knees (altering) (26), Knees Bending (28), Rowing (31) and Jump rope (8) show a small performance improvement by overlapping windowing. As for KNN (Figure 4.7(b)), performance reductions resulting from the use of overlapping windowing are higher than for DT. As an example, using overlapping drops the performance of activity Repetitive forward stretching (17) by 11%. This may be due to the nature of KNN (Cover et al., 1967), for which a small change in the dataset may have a large impact on the performance. Similar to DT, the performance for some activities also improves when overlapping windows are used, but by less than 2%. As can be seen in Figure 4.7(c) and Figure 4.7(d), the performance of all activities for NB and NCC are almost the same.

Dataset 2. Similar to Dataset 1, the activity-specific F1-score distributions obtained for this dataset



(a) FS1

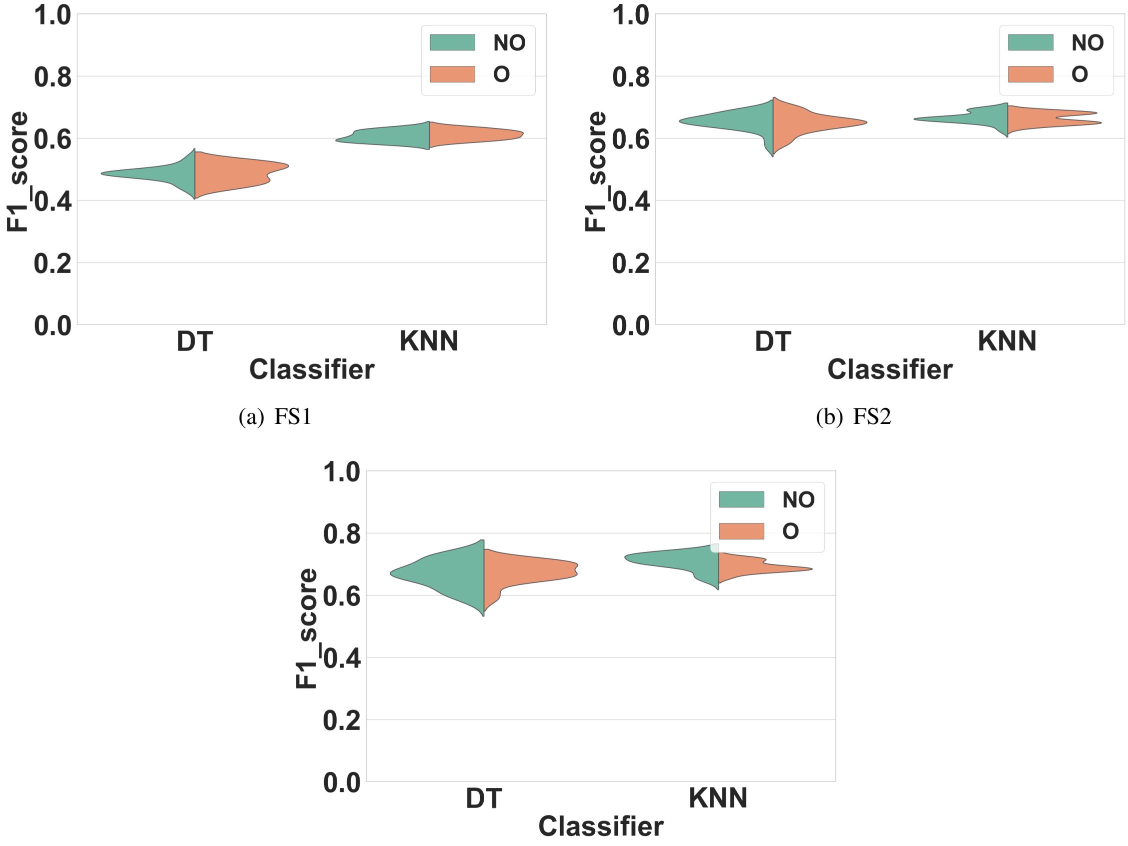
(b) FS2

(c) FS3

Figure 4.5: Experiment 3 – Subject-independent CV – Dataset 1.

for the two scenarios are similar. For this dataset, using overlapping windowing with KNN and DT slightly enhances the F1-score of the majority of activities in comparison to non-overlapping windowing, by 4% on average. The highest improvements are observed for activities Lawnmower (right) (21), Lateral Raise (19) and Seated Back Fly (31) for DT and Wall Ball (40), Seated Back Fly (31) and Power Boat pose (26) for KNN. Similar to Dataset 1, the performance of NB and NCC for all activities in both scenarios are almost the same.

This experiment shows that using overlapping windowing with subject-independent validation can impact the recognition performance of HAR systems for diverse activities differently. In spite of being mostly minor, for the activities in this study, overlapping windowing reduces the recognition accuracy of most activities and only a few of them show improvement. Moreover, the impact of overlapping windowing may be subject to the dataset, i.e., using overlapping windowing may impact



(a) FS1

(b) FS2

(c) FS3

Figure 4.6: Experiment 3 – Subject-independent CV – Dataset 2.

the performance of the system in recognizing a single activity in a different way. Running is a good example here. Using overlapping windowing reduces the F1-score of the system for this activity in Dataset 1, but it improves that in Dataset 2.

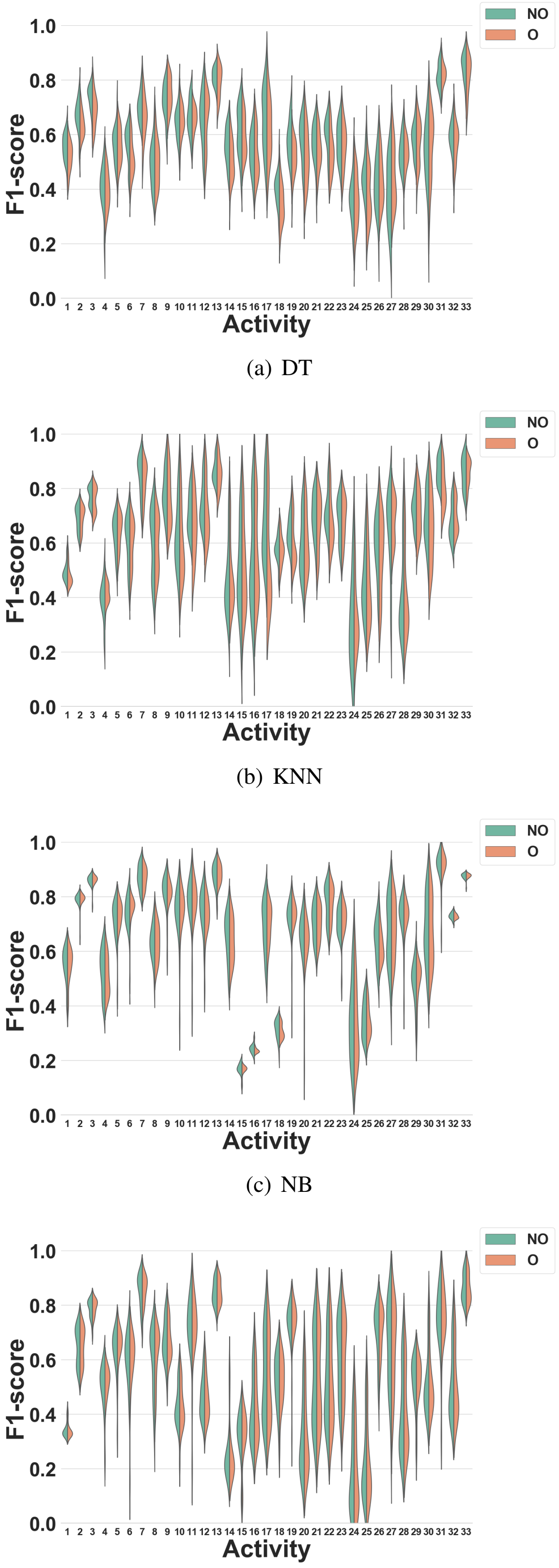
In summary, the recognition accuracy for most of the activities investigated in this study is quite comparable between the two scenarios with overlapping and non-overlapping windows.

Experiment 5: More discriminative features and neural-network classifier

In this experiment, we evaluate the effectiveness of non-overlapping windows using more discriminative time-domain features and a custom fine-tuned classifier. Namely, we target the HAR framework presented in Sarbishei (2019). The approach in Sarbishei (2019) utilizes configurable time-domain histogram features and a single hidden layer neural network classifier. It has been shown that the framework can outperform KNN classifiers as well as many other time-domain features explored by previous work. The framework in Sarbishei (2019) allows for a segmentation with a configurable sliding window. It presents results on subject independent CV using Dataset 2 and a 5s overlapping window sliding at 200ms steps. Here, we conduct a similar experiment, but with a non-overlapping window size of 5s. To provide a fair comparison, we use the same experimental setup compared to Sarbishei (2019), which is presented next.

We use time-domain histogram features, i.e., 120 bins in total, where 20 bins are assigned uniformly to each individual sensor axis. The accelerometer full-scale range has been set to $\pm 2g$, while the gyroscope full-scale range has been set to $\pm 512dps$. The single-hidden-layer neural network classifier consists of 120 input neurons, 60 hidden neurons and 7 output neurons targeting 6 activities and one noise class representing all other activities and the no activity periods. The Sigmoid (Softmax) trigger function has been used for the hidden (output) layer. Training the neural network has been done in Tensorflow 1.12.0 using a batch size of 32, and the AdamOptimizer algorithm. The number of epochs is chosen, such that the overall number of data points that are fed to the neural network for training becomes identical compared to the experiment in Sarbishei (2019), i.e., similar training time.

The normalized confusion matrix for both overlapping and non-overlapping windows under a subject-independent CV process is shown in Table 4.3. The rows refer to the true activities, while the columns correspond to the predicted ones. Each entry in the confusion matrix has two values, where the top value refers to the scenario with overlapping windows, while the bottom value corresponds to the scenario with non-overlapping windows. The results indicate that the use of overlapping windows provides minor improvements on recognition accuracy compared to the non-overlapping windows under subject independent cross validation, even when discriminative features and a well-trained neural network classifier are utilized.



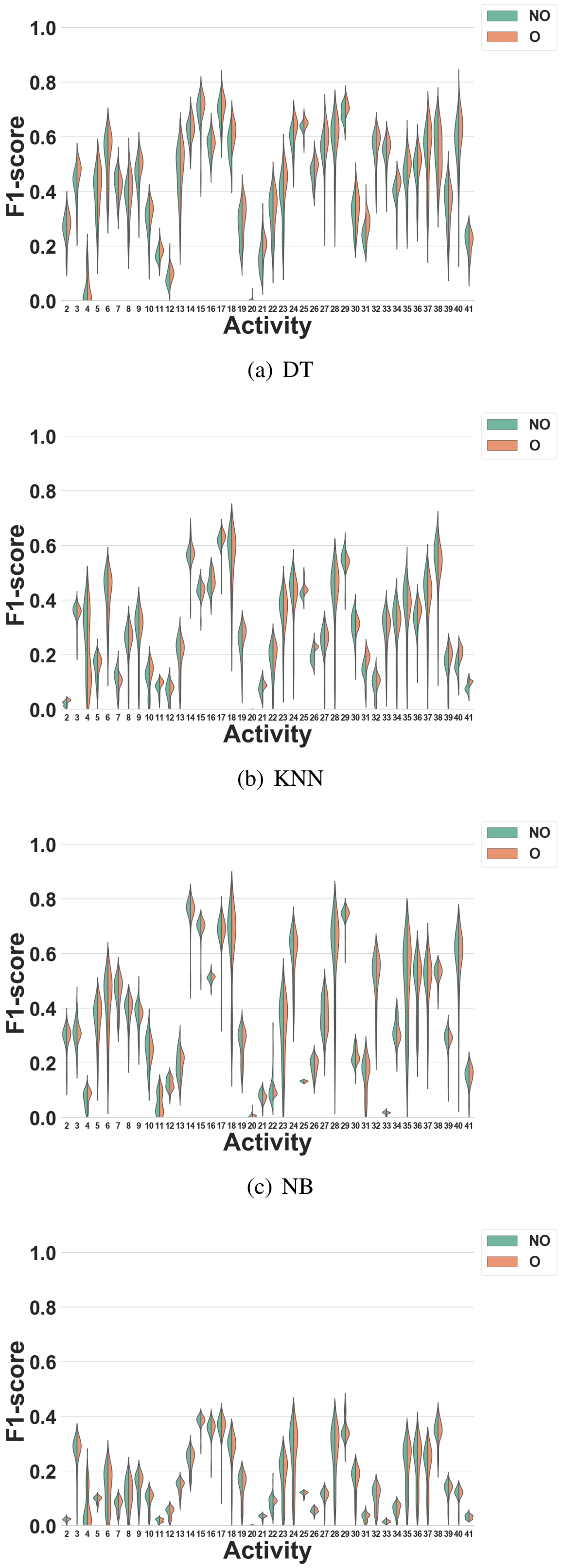
(a) DT

(b) KNN

(c) NB

(d) NCC

Figure 4.7: Experiment 4 – Subject-independent CV – Dataset 1.



(a) DT

(b) KNN

(c) NB

(d) NCC

Figure 4.8: Experiment 4 – Subject-independent CV – Dataset 2.

Activities	Noise (Others)	Curl	Triceps	Run	Elliptical	JumpJacks	Kettlebell
Noise (Others)	**0.99407** **0.9928**	0.00094 0.00103	0.00091 0.00099	0.00209 0.00286	0.00127 0.00129	0.00026 0.00028	0.00045 0.00074
Curl	0.1426 0.1369	**0.85082** **0.8492**	0.00659 0.0139	0 0	0 0	0 0	0 0
Triceps	0.17235 0.19643	0.00451 0.00487	**0.82315** **0.7987**	0 0	0 0	0 0	0 0
Run	0.20903 0.20127	0 0	0 0	**0.78668** **0.79237**	0.00429 0.00565	0 0	0 0.00071
Elliptical	0.16742 0.16265	0 0	0 0	0.00916 0.03113	**0.82342** **0.80622**	0 0	0 0
JumpJacks	0.20147 0.23729	0 0	0 0	0 0	0 0	**0.79853** **0.76271**	0 0
Kettlebell	0.18938 0.21484	0.00287 0	0 0	0 0	0 0	0 0	**0.80775** **0.78516**

Table 4.3: Normalized confusion matrix for Experiment 5, i.e., six activities and one noise class representing all other activities and no activity periods from Dataset 2 under subject independent cross validation. Rows are true activities, and columns are predicted ones. Each entry has two values, where the top (bottom) value refers to the scenario with overlapping (non-overlapping) windows.

4.3 Summary

In this chapter, we show the results of five different experiments about the impact of using overlapping and nonoverlapping windowing techniques with two different system evaluation methods. In general, the results show that using nonoverlapping windowing, we can achieve the same activity recognition performance. Moreover, the results of Experiments 3 and 5 confirm that our hypothesis is true for different hyperparameters and classifiers. In the next chapter, we discuss the results and conclude the thesis by summarizing our work and offering remarks on possible future works.

Chapter 5

Discussion and conclusion

5.1 Discussion

As can be seen by comparing the results of Experiment 1 and 2 for non-overlapping windowing, using Subject independent CV instead of Subject-dependent CV reduces the F1-score of KNN and DT by 10% and 11% on average for Datasets 1 and 2 respectively, which is substantial. It confirms that samples drawn from the same subject cannot be considered independent. In an ARP setup, Subject dependent CV overestimates the classification performance and should, therefore, be avoided. The detrimental effect of subject-dependent CV is even larger when overlapping time windows are used. In this case, as can be seen by comparing the results of Experiments 1 and 2 for overlapping windowing, Subject-independent CV reduces the F1-score of KNN and DT by 16% and 21% on average for Dataset 1 and 2 respectively. This further confirms that within-subject dependencies between time windows account for a significant part of the performance measured through Subject-dependent CV. Furthermore, for overlapping windows, the performance difference between subject-dependent CV and subject independent CV increases with the window size. This is consistent with our previous comments, as the amount of overlap between overlapping windows, and therefore their correlation, also increases with the window size.

Comparing the results for overlapping and non-overlapping windowing in Figure 4.1 and Figure 4.2 shows that when using Subject-dependent CV, overlapping windowing can improve the recognition power of the classifiers, which coincides with the general idea that overlapping windowing improves the performance of HAR systems (Janidarmian et al., 2014, 2017). However, our results confirm that such improvement comes from the increasing correlation among test and train folds due to the underlying problems of Subject-dependent CV in HAR systems. In contrast, when using Subject independent CV, the impact of using overlapping windows is minor to negligible, as can be seen in Figure 4.3 and Figure 4.4. This is in contradiction with the common argument that overlapping windows improve classification performance by bringing more data to the classifier. However, it also confirms our hypothesis that the performance increase coming from overlapping windows is, in fact, coming from the extra correlation between time windows, when Subject-dependent CV is used.

Experiment 3 showed that this conclusion also holds when using a different set of hyperparameters, which improves the generalizability of our result.

The results of Experiment 4 show that the impact of overlapping windowing with subject independent CV can be different per activity. In other words, overlapping windowing for some activities such as Trunk Twist and Lateral Raise improves the recognition performances and for others like Repetitive forward stretching and Heels not. However, such changes remain negligible for most activities and using this technique in HAR seems to be non-beneficial.

Experiment 5 explored the use of more discriminative features with a neural-network model, and the results similarly suggest that the use of overlapping windows does not provide major performance improvements.

Finally, Table 5.1 shows the data size and required time for segmentation and training in overlapping and non-overlapping windowing techniques with subject independent CV for two datasets. Segmenting using overlapping windows is almost twice longer than with non-overlapping windows, which is significant. Similarly, training on the data windowed by overlapping windows technique takes 4 times of non-overlapping one. As for storage, the size of segmented data by overlapping sliding windows technique is almost 9 times of data produced by non-overlapping one for both datasets. In spite of such increase in size and computation, this technique does not improve the performance of the classifiers when used with Subject independent CV.

Dataset	Raw size (GB)	Nonoverlapping windowing			Overlapping windowing		
		Segmentation		Training time (day)	Segmentation		Training time (day)
-	-	Time (Hour)	Size(GB)		Time (Hour)	Size (GB)	
1	2.4	6.0	2.3	1.0	11.0	21	4.0
2	3.4	12.0	5.8	2.0	20.0	51	8.0

Table 5.1: Overlapping windowing vs. nonoverlapping windowing required resources - Subject independent CV

5.2 Conclusion

We conclude that the suggested use of overlapping sliding windows in HAR systems is associated with underlying limitations of subject-dependent CV. When subject-independent CV is used, overlapping sliding windows do not improve the performance of HAR systems but still require substantially more resources than non-overlapping windows.

Our results show that the performance of all classifiers drops when subject-independent CV is used rather than subject-dependent CV. One possible way to address this problem would be to use features that are more common among subjects with different characteristics. Thus, in our future work, we will design such features and investigate their impact on the performance of HAR systems. This would enable building more generalized systems with a limited number of subjects.

References

Ahad, M. A. R., Tan, J., Kim, H., & Ishikawa, S. (2008). Human activity recognition: Various paradigms. In *2008 international conference on control, automation and systems* (pp. 1896–1901).

Al Machot, F., Elmachot, A., Ali, M., Al Machot, E., & Kyamakya, K. (2019). A deep-learning model for subject-independent human emotion recognition using electrodermal activity sensors. *Sensors*, *19*(7), 1659.

Altun, K., & Barshan, B. (2010). Human activity recognition using inertial/magnetic sensor units. In *International workshop on human behavior understanding* (pp. 38–51).

Amft, O., & Tröster, G. (2008). Recognition of dietary activity events using on-body sensors. *Artificial intelligence in medicine*, *42*(2), 121–136.

Aminian, K., Najafi, B., Büla, C., Leyvraz, P.-F., & Robert, P. (2002). Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes. *Journal of biomechanics*, *35*(5), 689–699.

Aminian, K., Rezakhanlou, K., De Andres, E., Fritsch, C., Leyvraz, P.-F., & Robert, P. (1999). Temporal feature estimation during walking using miniature accelerometers: an analysis of gait improvement after hip arthroplasty. *Medical & biological engineering & computing*, *37*(6), 686–691.

Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, 40–79.

Atallah, L., Lo, B., King, R., & Yang, G.-Z. (2011). Sensor positioning for activity recognition using wearable accelerometers. *IEEE transactions on biomedical circuits and systems*, *5*(4), 320–329.

Aung, M. S., Thies, S. B., Kenney, L. P., Howard, D., Selles, R. W., Findlow, A. H., & Goulermas, J. Y. (2013). Automated detection of instantaneous gait events using time frequency analysis and manifold embedding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *21*(6), 908–916.

Baños, O., Damas, M., Pomares, H., Rojas, I., Tóth, M. A., & Amft, O. (2012). A benchmark dataset to evaluate sensor displacement in activity recognition. In *Proceedings of the 2012 acm conference on ubiquitous computing* (pp. 1026–1035).

Banos, O., Galvez, J.-M., Damas, M., Pomares, H., & Rojas, I. (2014). Window size impact in human activity recognition. *Sensors*, *14*(4), 6474–6499.

Bao, L., & Intille, S. S. (2004a). Activity recognition from user-annotated acceleration data. In *International conference on pervasive computing* (pp. 1–17).

Bao, L., & Intille, S. S. (2004b). Activity recognition from user-annotated acceleration data. In *Pervasive.*

Benocci, M., Bächlin, M., Farella, E., Roggen, D., Benini, L., & Tröster, G. (2010). Wearable assistant for load monitoring: Recognition of on—body load placement from gait alterations. In *2010 4th international conference on pervasive computing technologies for healthcare* (pp. 1–8).

Bianchi, F., Redmond, S. J., Narayanan, M. R., Cerutti, S., & Lovell, N. H. (2010). Barometric pressure and triaxial accelerometry-based falls event detection. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *18*(6), 619–627.

Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, *46*(3), 33.

Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., & Kasturi, R. (2009). Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE transactions on intelligent transportation systems*, *11*(1), 206–224.

Chen, K.-Y., Harniss, M., Patel, S., & Johnson, K. (2014). Implementing technology-based embedded assessment in the home and community life of individuals aging with disabilities: A participatory research and development study. *Disability and Rehabilitation: Assistive Technology*, *9*(2), 112–120.

Chen, Y.-P., Yang, J.-Y., Liou, S.-N., Lee, G.-Y., & Wang, J.-S. (2008). Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Applied Mathematics and Computation*, *205*(2), 849–860.

Cheng, J., Amft, O., & Lukowicz, P. (2010). Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In *International conference on pervasive computing* (pp. 319–336).

Choujaa, D., & Dulay, N. (2008). Tracme: Temporal activity recognition using mobile phone data. In *2008 ieee/ifip international conference on embedded and ubiquitous computing* (Vol. 1, pp. 119–126).

Coggeshall, S., & Wu, G. (2005). Asset allocation and long-term returns: An empirical approach.

Consolvo, S., McDonald, D. W., Toscos, T., Chen, M. Y., Froehlich, J., Harrison, B., … others (2008). Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1797–1806).

Cortina-Puig, M., Istamboulie, G., Noguer, T., & Marty, J.-L. (2010). Analysis of pesticide mixtures using intelligent biosensors. *Intelligent and Biosensors*, 205–216.

Cover, T. M., Hart, P., et al. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Dehghani, A., Glatard, T., & Shihab, E. (2019). *Subject cross validation in human activity recognition.*

Dehghani, A., Sarbishei, O., Glatard, T., & Shihab, E. (2019). A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors. *Sensors*, *19*(22). Retrieved from https://www.mdpi.com/1424-8220/19/22/5026

Dernbach, S., Das, B., Krishnan, N. C., Thomas, B. L., & Cook, D. J. (2012). Simple and complex activity recognition through smart phones. In *2012 eighth international conference on intelligent environments* (pp. 214–221).

Dobkin, B. H., Xu, X., Batalin, M., Thomas, S., & Kaiser, W. (2011). Reliability and validity of bilateral ankle accelerometer algorithms for activity recognition and walking speed after stroke. *Stroke*, *42*(8), 2246–2250.

Ermes, M., Parkka, J., & Cluitmans, L. (2008, Aug). Advancing from offline to online activity recognition with wearable sensors. In *2008 30th annual international conference of the ieee engineering in medicine and biology society* (p. 4451-4454). doi: 10.1109/IEMBS.2008.4650199

Figo, D., Diniz, P. C., Ferreira, D. R., & Cardoso, J. M. (2010). Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, *14*(7), 645–662.

Gjoreski, H., & Gams, M. (2011). Accelerometer data preparation for activity recognition. In *Proceedings of the international multiconference information society, ljubljana, slovenia* (Vol. 1014, p. 1014).

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, *5*(6), 989–993.

Han, C. W., Kang, S. J., & Kim, N. S. (2010). Implementation of hmm-based human activity recognition using single triaxial accelerometer. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, *93*(7), 1379–1383.

Hanai, Y., Nishimura, J., & Kuroda, T. (2009). Haar-like filtering for human activity recognition using 3d accelerometer. In *2009 ieee 13th digital signal processing workshop and 5th ieee signal processing education workshop* (pp. 675–678).

He, Z., & Jin, L. (2009). Activity recognition from acceleration data based on discrete consine transform and svm. In *2009 ieee international conference on systems, man and cybernetics* (pp. 5041–5044).

Hemalatha, C. S., & Vaidehi, V. (2013). Frequent bit pattern mining over tri-axial accelerometer data streams for recognizing human activities and detecting fall. *Procedia Computer Science*, *19*, 56–63.

Huynh, T., Blanke, U., & Schiele, B. (2007). Scalable recognition of daily activities with wearable sensors. In *Loca.*

Huynh, T., & Schiele, B. (2005). Analyzing features for activity recognition. In *Proceedings of the*

2005 joint conference on smart objects and ambient intelligence: innovative context-aware services: usages and technologies (pp. 159–163).

Iglesias, J., Cano, J., Bernardos, A. M., & Casar, J. R. (2011). A ubiquitous activity-monitor to prevent sedentariness. In *2011 ieee international conference on pervasive computing and communications workshops (percom workshops)* (pp. 319–321).

Janidarmian, M., Radecka, K., & Zilic, Z. (2014). Automated diagnosis of knee pathology using sensory data. In *2014 4th international conference on wireless mobile communication and healthcare-transforming healthcare through innovations in mobile and wireless technologies (mobihealth)* (pp. 95–98).

Janidarmian, M., Roshan Fekr, A., Radecka, K., & Zilic, Z. (2017). A comprehensive analysis on wearable acceleration sensors in human activity recognition. *Sensors*, *17*(3), 529.

Jasiewicz, J. M., Allum, J. H., Middleton, J. W., Barriskill, A., Condie, P., Purcell, B., & Li, R. C. T. (2006). Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals. *Gait & posture*, *24*(4), 502–509.

Jatoba, L. C., Grossmann, U., Kunze, C., Ottenbacher, J., & Stork, W. (2008). Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *2008 30th annual international conference of the ieee engineering in medicine and biology society* (pp. 5250–5253).

Jatoba, L. C., Grossmann, U., Kunze, C., Ottenbacher, J., & Stork, W. (2008, Aug). Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *2008 30th annual international conference of the ieee engineering in medicine and biology society* (p. 5250-5253). doi: 10.1109/IEMBS.2008.4650398

Jiang, M., Shang, H., Wang, Z., Li, H., & Wang, Y. (2011). A method to deal with installation errors of wearable accelerometers for human activity recognition. *Physiological measurement*, *32*(3), 347.

Joseph, C., Kokulakumaran, S., Srijeyanthan, K., Thusyanthan, A., Gunasekara, C., & Gamage, C. (2010). A framework for whole-body gesture recognition from video feeds. In *2010 5th international conference on industrial and information systems* (pp. 430–435).

Kao, T.-P., Lin, C.-W., & Wang, J.-S. (2009). Development of a portable activity detector for daily activity recognition. In *2009 ieee international symposium on industrial electronics* (pp. 115–120).

Kasteren, T., Englebienne, G., & Kröse, B. (2010). An activity monitoring system for elderly care using generative and discriminative models. *Personal and ubiquitous computing*, *14*(6), 489–498.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2001). An online algorithm for segmenting time series. In *Proceedings 2001 ieee international conference on data mining* (pp. 289–296).

Kern, N., Schiele, B., & Schmidt, A. (2003). Multi-sensor activity context detection for wearable computing. In *European symposium on ambient intelligence* (pp. 220–232).

Khan, A. M., Lee, Y.-K., Lee, S.-Y., & Kim, T.-S. (2010). Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *2010 5th international conference on future information technology* (pp. 1–6).

Krause, A., Siewiorek, D. P., Smailagic, A., & Farringdon, J. (2003). Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *Iswc* (Vol. 3, p. 88).

Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, *12*(2), 74–82.

Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, *15*(3), 1192–1209.

Lara, O. D., Pérez, A. J., Labrador, M. A., & Posada, J. D. (2012). Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and mobile computing*, *8*(5), 717–729.

Lara, D., & Labrador, M. A. (2012, Jan). A mobile platform for real-time human activity recognition. In *2012 ieee consumer communications and networking conference (ccnc)* (p. 667-671). doi: 10.1109/CCNC.2012.6181018

Lee, Y.-S., & Cho, S.-B. (2011). Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer. In *International conference on hybrid artificial intelligence systems* (pp. 460–467).

Lester, J., Choudhury, T., & Borriello, G. (2006). A practical approach to recognizing physical activities. In *International conference on pervasive computing* (pp. 1–16).

Levner, I. (2005). Feature selection and nearest centroid classification for protein mass spectrometry. *BMC bioinformatics*, *6*(1), 68.

Lin, J. J., Mamykina, L., Lindtner, S., Delajoux, G., & Strub, H. B. (2006). Fish'n'steps: Encouraging physical activity with an interactive computer game. In *International conference on ubiquitous computing* (pp. 261–278).

Luštrek, M., & Kaluža, B. (2009). Fall detection and activity recognition with machine learning. *Informatica*, *33*(2).

Mannini, A., Intille, S. S., Rosenberger, M., Sabatini, A. M., & Haskell, W. (2013). Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and science in sports and exercise*, *45*(11), 2193.

Mantyjarvi, J., Himberg, J., & Seppanen, T. (2001). Recognizing human motion with multiple acceleration sensors. In *2001 ieee international conference on systems, man and cybernetics. e-systems and e-man for cybernetics in cyberspace (cat. no. 01ch37236)* (Vol. 2, pp. 747–752).

Marx, R. (2012). Ad-hoc accelerometer activity recognition in the iball. In *Proceedings of the 2012 acm conference on ubiquitous computing, pittsburgh, pa, usa* (Vol. 58).

Maurer, U., Smailagic, A., Siewiorek, D., & Deisher, M. (2006b, 05). Activity recognition and monitoring using multiple sensors on different body positions. In (Vol. 2006, p. 4 pp.-). doi: 10.1109/BSN.2006.6

Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. (2006a). *Activity recognition and monitoring using multiple sensors on different body positions* (Tech. Rep.). CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

Minnen, D., Westeyn, T., Ashbrook, D., Presti, P., & Starner, T. (2007). Recognizing soldier activities in the field. In *4th international workshop on wearable and implantable body sensor networks (bsn 2007)* (pp. 236–241).

Miscellaneous clustering methods. (2011). In *Cluster analysis* (p. 215-255). John Wiley Sons, Ltd. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470977811.ch8 doi: 10.1002/9780470977811.ch8

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, *6*(4), 525–533.

Morris, D., Saponas, T. S., Guillory, A., & Kelner, I. (2014). Recofit: using a wearable sensor to find, recognize, and count repetitive exercises. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 3225–3234).

Najafi, B., Aminian, K., Paraschiv-Ionescu, A., Loew, F., Bula, C. J., & Robert, P. (2003). Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on biomedical Engineering*, *50*(6), 711–723.

Nam, Y., & Park, J. W. (2013a). Child activity recognition based on cooperative fusion model of a triaxial accelerometer and a barometric pressure sensor. *IEEE journal of biomedical and health informatics*, *17*(2), 420–426.

Nam, Y., & Park, J. W. (2013b). Physical activity recognition using a single triaxial accelerometer and a barometric sensor for baby and child care in a home environment. *Journal of Ambient Intelligence and Smart Environments*, *5*(4), 381–402.

Nyan, M., Tay, F., Seah, K., & Sitoh, Y. (2006). Classification of gait patterns in the time–frequency domain. *Journal of biomechanics*, *39*(14), 2647–2656.

Olszewski, R. (2001, 03). Generalized feature extraction for structural pattern recognition in time-series data.

Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J., & Korhonen, I. (2006). Activity classification using realistic data from wearable sensors. *IEEE Transactions on information technology in biomedicine*, *10*(1), 119–128.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Perez, A. J., Labrador, M. A., & Barbeau, S. J. (2010). G-sense: a scalable architecture for global sensing and monitoring. *IEEE Network*, *24*(4), 57–64.

Pirttikangas, S., Fujinami, K., & Nakajima, T. (2006). Feature selection and activity recognition from wearable sensors. In *International symposium on ubiquitious computing systems* (pp. 516–527).

Preece, S. J., Goulermas, J. Y., Kenney, L. P., & Howard, D. (2008). A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, *56*(3), 871–879.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005). Activity recognition from accelerometer data. In *Aaai* (Vol. 5, pp. 1541–1546).

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, *6*(2), 13.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proceedings of the ieee international conference on neural networks* (Vol. 1993, pp. 586–591).

Sarbishei, O. (2019). A platform and methodology enabling real-time motion pattern recognition on low-power smart devices. In *World forum on internet of things* (pp. 269–272). IEEE.

Sarbishei, O. (2019, April). A platform and methodology enabling real-time motion pattern recognition on low-power smart devices. In *2019 ieee 5th world forum on internet of things (wf-iot)* (p. 269-272). doi: 10.1109/WF-IoT.2019.8767219

Sarkar, J., Lee, Y.-K., Lee, S., et al. (2011). Gpars: A general-purpose activity recognition system. *Applied Intelligence*, *35*(2), 242–259.

Sazonov, E., Metcalfe, K., Lopez-Meyer, P., & Tiffany, S. (2011). Rf hand gesture sensor for monitoring of cigarette smoking. In *2011 fifth international conference on sensing technology* (pp. 426–430).

Sazonov, E. S., Makeyev, O., Schuckers, S., Lopez-Meyer, P., Melanson, E. L., & Neuman, M. R. (2009). Automatic detection of swallowing events by acoustical means for applications of monitoring of ingestive behavior. *IEEE Transactions on Biomedical Engineering*, *57*(3), 626–633.

Sekine, M., Tamura, T., Togawa, T., & Fukui, Y. (2000). Classification of waist-acceleration signals in a continuous walking record. *Medical engineering & physics*, *22*(4), 285–291.

Selles, R. W., Formanoy, M. A., Bussmann, J. B., Janssens, P. J., & Stam, H. J. (2005). Automated estimation of initial and terminal contact timing using accelerometers; development and validation in transtibial amputees and controls. *IEEE transactions on neural systems and rehabilitation engineering*, *13*(1), 81–88.

Sharma, A., Lee, Y.-D., & Chung, W.-Y. (2008). High accuracy human activity monitoring using neural network. In *2008 third international conference on convergence and hybrid information technology* (Vol. 1, pp. 430–435).

Sharma, A., & Paliwal, K. (2010). Improved nearest centroid classifier with shrunken distance measure for null lda method on cancer classification problem. *Electronics letters*, *46*(18), 1251–1252.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., … Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *Cvpr 2011* (pp. 1297–1304).

Siirtola, P., & Röning, J. (2012). User-independent human activity recognition using a mobile phone: Offline recognition vs. real-time on device recognition. In *Distributed computing and artificial intelligence* (pp. 617–627). Springer.

Stikic, M., Huynh, T., Van Laerhoven, K., & Schiele, B. (2008). Adl recognition based on the combination of rfid and accelerometer sensing. In *2008 second international conference on pervasive computing technologies for healthcare* (pp. 258–263).

Sun, L., Zhang, D., Li, B., Guo, B., & Li, S. (2010). Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In *International conference on ubiquitous intelligence and computing* (pp. 548–562).

Suutala, J., Pirttikangas, S., & Röning, J. (2007). Discriminative temporal smoothing for activity recognition from wearable sensors. In *International symposium on ubiquitious computing systems* (pp. 182–195).

Tamura, T., Yoshimura, T., Sekine, M., Uchida, M., & Tanaka, O. (2009). A wearable airbag to prevent fall injuries. *IEEE Transactions on Information Technology in Biomedicine*, *13*(6), 910–914.

Tan, S. (2008). An improved centroid classifier for text categorization. *Expert Systems with Applications*, *35*(1-2), 279–285.

Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A., & Friedman, R. (2007). Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart monitor. In *In: Proc. int. symp. on wearable comp.*

Theodoridis, S., & Koutroumbas, K. (1998). Pattern recognition.

Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., et al. (2003). Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, *18*(1), 104–117.

Tolstikov, A., Hong, X., Biswas, J., Nugent, C., Chen, L., & Parente, G. (2011). Comparison of fusion methods based on dst and dbn in human activity recognition. *Journal of Control Theory and Applications*, *9*(1), 18–27.

Trevor, H., Robert, T., & JH, F. (2009). *The elements of statistical learning: data mining, inference, and prediction.* New York, NY: Springer.

Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology*, *18*(11), 1473.

Vinh, L. T., Lee, S., Le, H. X., Ngo, H. Q., Kim, H. I., Han, M., & Lee, Y.-K. (2011, October). Semi-markov conditional random fields for accelerometer-based activity recognition. *Applied Intelligence*, *35*(2), 226–241. Retrieved from http://dx.doi.org/10.1007/s10489-010-0216-5 doi: 10.1007/s10489-010-0216-5

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2018). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3-11.

Wang, J.-H., Ding, J.-J., Chen, Y., & Chen, H.-H. (2012). Real time accelerometer-based gait recognition using adaptive windowed wavelet transforms. In *2012 ieee asia pacific conference on circuits and systems* (pp. 591–594).

Wang, N., Ambikairajah, E., Lovell, N. H., & Celler, B. G. (2007). Accelerometry based classification of walking patterns using time-frequency analysis. In *2007 29th annual international conference of the ieee engineering in medicine and biology society* (pp. 4899–4902).

Yang, J., Lee, J., & Choi, J. (2011). Activity recognition based on rfid object usage for smart mobile devices. *Journal of Computer Science and Technology*, *26*(2), 239–246.

Yoshizawa, M., Takasaki, W., & Ohmura, R. (2013). Parameter exploration for response time reduction in accelerometer-based activity recognition. In *Proceedings of the 2013 acm conference on pervasive and ubiquitous computing adjunct publication* (pp. 653–664).

Zambon, M., Lawrence, R., Bunn, A., & Powell, S. (2006). Effect of alternative splitting rules on image processing using classification tree analysis. *Photogrammetric Engineering & Remote Sensing*, *72*(1), 25–30.

Zhang, M., & Sawchuk, A. A. (2011). A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *Proceedings of the 6th international conference on body area networks* (pp. 92–98).

Zhen-Yu He, & Lian-Wen Jin. (2008, July). Activity recognition from acceleration data using ar model representation and svm. In *2008 international conference on machine learning and cybernetics* (Vol. 4, p. 2245-2250). doi: 10.1109/ICMLC.2008.4620779

Zheng, Y., Wong, W.-K., Guan, X., & Trost, S. (2013). Physical activity recognition from accelerometer data using a multi-scale ensemble method. In *Twenty-fifth iaai conference.*

Zhenyu He, Zhibin Liu, Lianwen Jin, Li-Xin Zhen, & Jian-Cheng Huang. (2008, Dec). Weightlessness feature — a novel feature for single tri-axial accelerometer based activity recognition. In *2008 19th international conference on pattern recognition* (p. 1-4). doi: 10.1109/ICPR.2008.4761688

Zhu, C., & Sheng, W. (2009, May). Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *2009 ieee international conference on robotics and automation* (p. 2154-2159). doi: 10.1109/ROBOT.2009.5152756