January 2012

On the Automatic Recognition of Human Activities using Heterogeneous Wearable Sensors

Oscar David Lara Yejas

University of South Florida, olarayej@mail.usf.edu

Follow this and additional works at: http://scholarcommons.usf.edu/etd

Part of the American Studies Commons, and the Computer Sciences Commons

Scholar Commons Citation
Graduate Theses and Dissertations.
http://scholarcommons.usf.edu/etd/4120

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
On the Automatic Recognition of Human Activities using Heterogeneous Wearable Sensors

by

Oscar D. Lara Yejas

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Miguel Labrador, Ph.D.
Wilfrido Moreno, Ph.D.
Rafael Perez, Ph.D.
Xiaoning Qian, Ph.D.
Kandethody Ramachandran, Ph.D.

Date of Approval:
June 15, 2012

Keywords: context-aware applications, machine learning, ubiquitous sensing, pervasive computing, human-centric sensing

Copyright © 2012, Oscar D. Lara Yejas
Acknowledgments

I would first like to thank my advisor and mentor, Dr. Miguel Labrador, for his constant support and motivation throughout my journey as a graduate student. His clever ideas and suggestions —after countless technical and non-technical discussions— have been very valuable for the fulfilment of this work. I have also had the opportunity to learn from his extensive academic, professional, and personal experience.

This work was partially funded by the National Science Foundation under grant No. 1062160, corresponding to the Research Experience for Undergraduates (REU) program at USF. Thanks to the REU students José Cadena, Elizabeth Rodríguez, and Aaron Young, who made significant contributions to the development of Vigilante, the mobile application for real-time activity recognition presented in Chapter 4. Furthermore, I am grateful to my colleagues and friends Idálides Vergara, Alfredo Pérez, Diego Méndez, and Luis G. Jaimes, for contributing to this dissertation with their feedback and ideas. Special thanks to José Posada for his technical work on feature extraction and his great support as a friend.

My dissertation and my entire career as a computer scientist would not have been possible without the inspiration from my parents and my sister. Thanks to them for believing in me during all these years!

Finally, thanks to Anamaria for her support, love, and patience; for reviewing this manuscript, and teaching me the very little I know of the language I am writing in.
Table of Contents

List of Tables v
List of Figures vii
Abstract ix

Chapter 1 Introduction 1
1.1 Human Activity Recognition Approaches 2
1.2 Human Activity Recognition with Wearable Sensors 3
1.3 Problem Statement 4
1.4 Contributions 6
1.5 Structure of the Dissertation 8

Chapter 2 Related Work 9
2.1 Design Issues 9
2.1.1 Definition of the Activity Set 9
2.1.2 Selection of Attributes and Sensors 10
2.1.2.1 Environmental Attributes 10
2.1.2.2 Acceleration 11
2.1.2.3 Location 11
2.1.2.4 Physiological Signals 12
2.1.3 Obtrusiveness 12
2.1.4 Data Collection Protocol 13
2.1.5 Recognition Performance 13
2.1.6 Energy Consumption 14
2.1.7 Processing 14
2.1.8 User Flexibility 15
2.2 Activity Recognition Methods 16
2.2.1 Feature Extraction 16
2.2.1.1 Acceleration 18
2.2.1.2 Environmental Variables 19
2.2.1.3 Physiological Signals 20
2.2.1.4 Selection of the Window Length 20
2.2.1.5 Feature Selection 21
2.2.2 Learning 21
2.2.2.1 Supervised Learning 22
2.2.2.2 Semi-supervised Learning 24
2.2.2.3 Evaluation Methodologies 24
List of Tables

Table 2.1 Types of activities recognized by state-of-the-art HAR systems. 10
Table 2.2 Summary of feature extraction methods for acceleration signals. 19
Table 2.3 Summary of feature extraction methods for environmental variables. 20
Table 2.4 Classification algorithms used by state-of-the-art human activity recognition systems. 23
Table 2.5 $5 \times 2$-fold cross validation 26
Table 2.6 List of abbreviations and acronyms. 31
Table 2.7 Summary of state-of-the-art in online human activity recognition systems. 32
Table 2.8 Summary of state-of-the-art in offline human activity recognition systems. 39
Table 3.1 Physical characteristics of the participants. 56
Table 3.2 List of features extracted in this work. 58
Table 3.3 Common functions implemented by structure detectors. 59
Table 3.4 Percentage classification accuracy given by the $5 \times 2$-fold cross validation on $D_{vs}$. 65
Table 3.5 Percentage classification accuracy given by the $5 \times 2$-fold cross validation on $D_{acc}$. 65
Table 3.6 Impact of physiological signals in the classification accuracy: precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure. 68
Table 3.7 Impact of transient features in the classification accuracy: precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure. 69
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8</td>
<td>Confusion matrix for the best classifier $ALR_{acc+tra}^{5s}$ after five iterations with different random seeds.</td>
<td>70</td>
</tr>
<tr>
<td>4.1</td>
<td>Selected features for mobile activity recognition.</td>
<td>79</td>
</tr>
<tr>
<td>4.2</td>
<td>Physical characteristics of the new participants.</td>
<td>83</td>
</tr>
<tr>
<td>4.3</td>
<td>Parameters to evaluate Weka classifiers in mobile devices.</td>
<td>84</td>
</tr>
<tr>
<td>4.4</td>
<td>Confusion matrix for Individual 1 (5 activities).</td>
<td>85</td>
</tr>
<tr>
<td>4.5</td>
<td>Confusion matrix for Individuals 2 and 3 (5 activities).</td>
<td>86</td>
</tr>
<tr>
<td>4.6</td>
<td>Confusion matrix for Individuals 2 and 4 with three activities</td>
<td>87</td>
</tr>
<tr>
<td>4.7</td>
<td>Estimated energy consumption after executing the application for three hours.</td>
<td>88</td>
</tr>
<tr>
<td>5.1</td>
<td>Highest probability fallacy</td>
<td>92</td>
</tr>
<tr>
<td>5.2</td>
<td>Precision vs. recall fallacy</td>
<td>93</td>
</tr>
<tr>
<td>5.3</td>
<td>Dataset specifications.</td>
<td>95</td>
</tr>
<tr>
<td>5.4</td>
<td>Percentage average accuracies</td>
<td>96</td>
</tr>
<tr>
<td>5.5</td>
<td>Correlation matrix for the soybean dataset.</td>
<td>98</td>
</tr>
<tr>
<td>5.6</td>
<td>$B$ matrix for the balance-scale dataset.</td>
<td>99</td>
</tr>
<tr>
<td>5.7</td>
<td>$B$ matrix for the vehicle dataset.</td>
<td>101</td>
</tr>
<tr>
<td>5.8</td>
<td>Accuracy of base classifiers and ensembles with different noise levels.</td>
<td>103</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>General data flow for training and testing HAR systems based on wearable sensors.</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Generic data acquisition architecture for human activity recognition.</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>An example of the mapping from the raw dataset to the feature dataset.</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Acceleration signals for different activities.</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>The iterations of a 10-fold cross validation.</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Taxonomy of human activity recognition systems.</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Acceleration signals and heart rate for the activities walking and ascending.</td>
<td>52</td>
</tr>
<tr>
<td>3.2</td>
<td>Centinela’s data flow</td>
<td>54</td>
</tr>
<tr>
<td>3.3</td>
<td>Data collection architecture.</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>Mobile application user interface [1].</td>
<td>55</td>
</tr>
<tr>
<td>3.5</td>
<td>Heart rate signal for walking (bold) and flipped signal (thin).</td>
<td>59</td>
</tr>
<tr>
<td>3.6</td>
<td>Calculation of the magnitude of change feature.</td>
<td>61</td>
</tr>
<tr>
<td>3.7</td>
<td>Impact of physiological signals in the classification accuracy: precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure.</td>
<td>67</td>
</tr>
<tr>
<td>4.1</td>
<td>System architecture.</td>
<td>75</td>
</tr>
<tr>
<td>4.2</td>
<td>Mobile application design</td>
<td>76</td>
</tr>
<tr>
<td>4.3</td>
<td>Simplified UML class diagram for the sensing component</td>
<td>78</td>
</tr>
<tr>
<td>4.4</td>
<td>Simplified UML class diagram for the classification component</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>Mobile application user interface</td>
<td>83</td>
</tr>
</tbody>
</table>
Figure 4.6  Classification model generated by the C4.5 algorithm with three activities  87

Figure 5.1  Overall accuracy of the ensembles varying the number of classifiers.  101

Figure 5.2  Overall accuracy of ALR and FP with different noise levels.  104
Abstract

Delivering accurate and opportune information on people’s activities and behaviors has become one of the most important tasks within pervasive computing. Its wide spectrum of potential applications in medical, entertainment, and tactical scenarios, motivates further research and development of new strategies to improve accuracy, pervasiveness, and efficiency.

This dissertation addresses the recognition of human activities (HAR) with wearable sensors in three main regards: In the first place, physiological signals have been incorporated as a new source of information to improve the recognition accuracy achieved by conventional approaches, which rely on accelerometer signals solely. A new HAR system, Centinela, was born from such concept, employing structural feature extraction along with classifier ensembles, and achieving over 95% of recognition accuracy.

In the second place, real time activity recognition was enabled by Vigilante, a mobile HAR framework under the Android™ platform. Providing immediate feedback on the user’s activities is especially beneficial in healthcare and military applications, which may require alert triggering or support of decision making. The evaluation demonstrates that Vigilante is energy efficient while maintaining high accuracy (i.e., up to 96.8%) and low response time. The system features MECLA, a mobile library for the evaluation of classification algorithms, which is also suitable for further machine learning applications.

Finally, the activity recognition accuracy is improved by two new strategies for decision fusion and selection in multiple classifier systems: the failure product and the precision-recall difference. The experimental analysis confirms that the presented methods are beneficial, not only for recognizing human activities, but also for many other classification problems.
Chapter 1: Introduction

During the past decade, there has been an exceptional development of microelectronics and computer systems, enabling sensors and mobile devices with unprecedented characteristics. Their high computational power, small size, and low cost allow people to interact with the devices as part of their daily living. That was the genesis of Ubiquitous Sensing, an active research area with the main purpose of extracting knowledge from the data acquired by pervasive sensors [2]. Particularly, the recognition of human activities has become a task of high interest within the field, especially for medical, military, and security applications. For instance, patients with diabetes, obesity, or heart disease are often required to follow a well defined exercise routine as part of their treatment [3]. Therefore, recognizing activities such as walking, running, or cycling becomes quite useful to provide feedback to the caregiver about the patient’s behavior. Likewise, patients with dementia and other mental pathologies could be monitored to detect abnormal activities and thereby prevent undesirable consequences [4]. An interactive game or simulator might also require information about which activity the user is performing in order to respond accordingly. Finally, in tactical scenarios, precise information on the soldiers’ activities along with their locations and health conditions, is highly beneficial for their performance and safety. Such information is also helpful to support decision making in both combat and training environments. Given the relevance of HAR and its applications, an overview of the most noticeable approaches is presented next.
1.1 Human Activity Recognition Approaches

The recognition of human activities has been approached in two different ways, namely using *external* and *wearable* sensors. In the former, the devices are fixed in predetermined points of interest, so the inference of activities entirely depends on the voluntary interaction of the users with the sensors. In the latter, the devices are attached to the user.

Intelligent homes [5, 6, 7, 8] are a typical example of external sensing. These systems are able to recognize fairly complex activities (e.g., eating, taking a shower, washing dishes, etc.) because they rely on data from a number of sensors placed in target objects which people are supposed to interact with (e.g., stove, faucet, and washing machine). Nonetheless, nothing can be done if the user is out of the reach of the sensors or they perform activities that do not require interaction with them. Additionally, the installation and maintenance of the sensors usually entail high costs.

Cameras have also been employed as external sensors for HAR. In fact, the recognition of activities and gestures from video sequences has been the focus of extensive research [9, 10, 11, 12]. This is especially suitable for security (e.g., intrusion detection) and interactive applications. A remarkable example, and also commercially available, is the *Kinect* game console [13] developed by Microsoft. It allows the user to interact with the game by means of gestures, without any controller device. Nevertheless, video sequences certainly have some issues in HAR. The first one is *privacy*, as not everyone is willing to be permanently monitored and recorded by cameras. The second one is *pervasiveness* because video recording devices are difficult to attach to target individuals in order to obtain images of their entire body during daily living activities. The monitored individuals should then stay within a perimeter defined by the position and the capabilities of the camera(s). The last issue would be *complexity*, since video processing techniques are computationally expensive, hindering a real time HAR system from being scalable. The aforementioned limitations motivate the use of wearable sensors in HAR. Most of the measured attributes are related to the user’s movement (e.g., using accelerometers or GPS), environmental variables (e.g., temperature and humidity), or physiological signals (e.g., heart rate or electrocardiogram).
1.2 Human Activity Recognition with Wearable Sensors

Similar to other machine learning applications, activity recognition requires two stages, i.e., training and testing —also called evaluation. Figure 1.1 illustrates the common phases involved in these two processes. The training stage initially requires a time series dataset of measured attributes from individuals performing each activity. The time series are divided into time windows to apply feature extraction in order to filter relevant information in the raw signals and define metrics to compare them. Later, learning methods are used to generate an activity recognition model from the dataset of extracted features. Likewise, for testing, data are collected during a time window and a feature vector —also called feature...
A generic data acquisition architecture for HAR systems was also identified, as shown in Figure 1.2. In the first place, wearable sensors are attached to the person’s body to measure attributes of interest such as motion [14], location [15], temperature [16], ECG [17], among others. These sensors should communicate with an integration device (ID), which can be a cellphone [18, 19], a PDA [16], a laptop [20, 17], or a customized embedded system [21, 22]. The main purpose of the ID is to preprocess the data received from the sensors and, in some cases, send them to an application server for real time monitoring, visualization, and/or analysis. The communication protocol might be UDP/IP or TCP/IP, according to the desired level of reliability.

Notice that all of these components are not necessarily implemented in every HAR system. In [23, 24, 25], the data are collected offline, so there is neither communication nor server processing. Other systems incorporate sensors within the ID [26, 27, 28], or carry out the inference process directly on it [29, 27].

1.3 Problem Statement

The data collected from wearable sensors are naturally indexed over the time dimension, which allows to define the human activity recognition problem as follows:
Definition 1 (Human Activity Recognition Problem (HARP)) Given a set $S = \{S_0, ..., S_{k-1}\}$ of $k$ time series, each one from a particular measured attribute, and all defined within time interval $I = [t_\alpha, t_\omega]$, the goal is to find a temporal partition $\langle I_0, ..., I_{r-1} \rangle$ of $I$, based on the data in $S$, and a set of labels representing the activity performed during each interval $I_j$ (e.g., sitting, walking, etc.). This implies that time intervals $I_j$ are consecutive, non-empty, non-overlapping, and such that $\bigcup_{j=0}^{r-1} I_j = I$.

This definition is valid assuming that activities are not simultaneous, i.e., a person does not walk and run at the same time. Note that the HARP is not feasible to be solved deterministically. The number of combinations of attribute values and activities can be very large—or even infinite—and finding transition points becomes hard as the duration of each activity is generally unknown. Therefore, machine learning tools are widely used to recognize activities. A relaxed version of the problem is then introduced, dividing the time series into fixed length time windows, as follows:

Definition 2 (Relaxed HAR problem) Given (1) a set $W = \{W_0, ..., W_{m-1}\}$ of $m$ equally sized time windows, totally or partially labeled, and such that each $W_i$ contains a set of time series $S_i = \{S_{i,0}, ..., S_{i,k-1}\}$ from each of the $k$ measured attributes, and (2) a set $A = \{a_0, ..., a_{n-1}\}$ of activity labels, the goal is to find a mapping function $f : S_i \rightarrow A$ that can be evaluated for all possible values of $S_i$, such that $f(S_i)$ is as similar as possible to the actual activity performed during $W_i$.

Notice this relaxation introduces some error to the model during transition windows, since a person might perform more than one activity within a single time window. However, the number of transitions is expected to be much smaller than the total number of time windows, which makes the relaxation error not significant for most of the applications.

Definition 2 specifies the very first problem addressed in this dissertation: building a learning model to recognize activities offline given a set of measured attributes. Yet, a number of real-world applications are compelled to also deliver immediate feedback on the user’s activities, especially in healthcare and military operations. Such task requires
coupling the sensors with an integration device (e.g., a cellphone), bringing about additional challenges as mobile devices are constrained in terms of computational power, memory, and energy. These limitations are particularly critical in activity recognition applications which entail high demands due to communication, feature extraction, classification, and transmission of large amounts of raw data. Moreover, current open source machine learning API’s such as Weka [30] and JDM [31] are neither designed nor optimized to run on mobile platforms.

Thus, the second problem to be addressed in the present dissertation is the mobile implementation of a human activity recognition system meeting response time and energy consumption requirements.

Finally, countless learning methods could be used in activity recognition. A noticeable approach is to rely not only on a single model, but combine the output of several learners in order to produce more accurate and diverse predictions. That is the main philosophy behind multiclassifier systems, which are shown to be effective, though they entail additional computational complexity. In such direction, a third focus of study in this dissertation attempts to formulate new strategies to effectively combine predictions from several learners. The problem is formally defined as follows:

**Definition 3** Given a classification problem with a feature space $\mathcal{X} \subseteq \mathbb{R}^n$ and a set of classes $\Omega = \{\omega_0, ..., \omega_{n-1}\}$, an instance $x \in \mathcal{X}$ to be classified, and a set of predictions $S = \{s_0, ..., s_{k-1}\}$ for $x$, from $k$ classifiers, the goal of a multiclassifier system is to return the correct label $\omega^*$ iff $\exists s_i \in S | \omega^* = s_i$.

### 1.4 Contributions

This dissertation features the following main contributions:

- A comprehensive literature review on human activity recognition with wearable sensors [32]. A new two-level taxonomy was proposed based on the learning approach and the response time. Then, the principal issues and challenges in HAR are discussed, as
well as the main solutions to each one of them. A qualitative analysis of twenty eight HAR systems is presented, providing the reader with guidelines to select the most appropriate approach to use in a specific case study. As a result of this literature review, the first survey paper in the field was compiled and published.

• *Enhancing activity recognition with physiological signals* [33]: It was shown that ambulation activities can be recognized more accurately by incorporating physiological signals (e.g., heart rate, respiration rate, and skin temperature, among others), besides the traditionally used acceleration signals. Only a few works have been reported in this matter [20] yet they were not successful because of the use of time-domain statistical features; these features do not describe the morphological interrelationship among vital sign data. Instead, to take advantage of physiological signals, this dissertation shows the application of structural feature extraction methods, such as polynomial regression and transient features, in conjunction with classifier ensembles.

• *Enabling mobile context-aware applications in real time* [34, 35]: Vigilante is presented as a new real-time human activity recognition system under the Android mobile platform. Implementing activity recognition on a mobile device is advantageous in terms of energy consumption (as raw data would not have to be sent to a server), robustness (because it would not depend on unreliable wireless communication links), and scalability (since the most complex computations would not have to be executed in the server). The evaluation demonstrates that Vigilante allows for significant energy savings while maintaining high accuracy and low response time.

• *Improving recognition accuracy with Multiple Classifier Systems* [36]. Two new strategies for decision selection and fusion in multiple classifier systems are presented. The evaluation of seven base classifiers and eleven datasets demonstrates that the proposed methods significantly improve the classification accuracy, not only for HAR but also for other machine learning problems. Moreover, a new algorithm is proposed to select base classifiers in order to maximize the benefits of such strategies.
1.5  Structure of the Dissertation

This dissertation is organized as follows: Chapter 2 covers the related work in HAR using wearable sensors and Multiclassifier Systems. It also features introductory material to pattern recognition techniques. Chapter 3 introduces Centinela as a HAR system considering physiological and acceleration signals. The system design, evaluation, and main results are also part of this chapter. Later, Chapter 4 is devoted to Vigilante, a mobile framework to enable real time activity recognition. Furthermore, the design of MECLA—the underlying library for mobile evaluation of classification algorithms—is illustrated. Chapter 5 presents two new strategies for decision fusion and selection in multiple classifier systems, and their application to human activity classification. Finally, Chapter 6 summarizes the main findings and contributions of this dissertation along with a number of ideas for further research consideration.
Chapter 2: Related Work

The first works on human activity recognition (HAR) date back to the late ’90s [37, 38]. However, there are still a number of opportunities for the development of new techniques to improve accuracy, efficiency, and pervasiveness. This chapter surveys the state of the art in two research areas: (1) HAR based on wearable sensors and (2) multiple classifier systems. First, Section 2.1 introduces the main design issues for recognizing activities and the most important solutions to each one of them. Section 2.2 describes the principal pattern recognition techniques applied in HAR, comprising feature extraction and learning methods. Section 2.3 shows a qualitative evaluation of state-of-the-art HAR systems. Finally, Section 2.4 gives an overview on the most relevant approaches in multiple classifier systems.

2.1 Design Issues

Any system attempting to recognize human activities is compelled to address at least eight main design issues: (1) definition of the activity set, (2) selection of attributes and sensors, (3) obtrusiveness, (4) data collection protocol, (5) recognition performance, (6) energy consumption, (7) processing, and (8) user flexibility. The main aspects and solutions related to each one of them are analyzed next. These design issues will be later referred to in Section 2.3 to qualitatively analyze the state-of-the-art HAR systems.

2.1.1 Definition of the Activity Set

The design of any HAR system depends on the activities to be recognized. In fact, changing the activity set $A$ immediately turns a given HARP into a completely different
Table 2.1: Types of activities recognized by state-of-the-art HAR systems.

<table>
<thead>
<tr>
<th>Group</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambulation</td>
<td>Walking, running, sitting, standing still, lying, climbing stairs,</td>
</tr>
<tr>
<td></td>
<td>descending stairs, riding escalator, and riding elevator.</td>
</tr>
<tr>
<td>Transportation</td>
<td>Riding a bus, cycling, and driving.</td>
</tr>
<tr>
<td>Phone usage</td>
<td>Text messaging, making a call.</td>
</tr>
<tr>
<td>Daily activities</td>
<td>Eating, drinking, working at the PC, watching TV, reading,</td>
</tr>
<tr>
<td></td>
<td>brushing teeth, stretching, scrubbing, and vacuuming.</td>
</tr>
<tr>
<td>Exercise/fitness</td>
<td>Rowing, lifting weights, spinning, Nordic walking, and doing</td>
</tr>
<tr>
<td></td>
<td>push ups.</td>
</tr>
<tr>
<td>Military</td>
<td>Crawling, kneeling, situation assessment, and opening a door.</td>
</tr>
<tr>
<td>Upper body</td>
<td>Chewing, speaking, swallowing, sighing, and moving the head.</td>
</tr>
</tbody>
</table>

problem. From the literature, seven groups of activities can be distinguished. These groups and the individual activities that belong to each group are summarized in Table 2.1.

The nature of the activities has a direct relation with the required sensors. For example, ambulation activities such as walking and running could be determined using accelerometers. Others such as talking might need a microphone whereas swallowing would require specialized sensors on the individual’s throat [39].

2.1.2 Selection of Attributes and Sensors

Four groups of attributes are measured using wearable sensors in a HAR context: environmental attributes, acceleration, location, and physiological signals.

2.1.2.1 Environmental Attributes

These attributes, such as temperature, humidity, audio level, etc., are intended to provide context information describing the individual’s surroundings. If the audio level and light intensity are fairly low, for instance, the subject may be sleeping. Various existing systems have utilized microphones, light sensors, humidity sensors, and thermometers, among others [22, 16]. Those sensors alone, though, might not provide sufficient information as individuals can perform each activity under diverse contextual conditions in terms
of weather, audio loudness, or illumination. Therefore, environmental sensors are generally accompanied by accelerometers and other sensors [4].

### 2.1.2.2 Acceleration

Triaxial accelerometers are perhaps the most broadly used sensors to recognize ambulation activities (e.g., walking, running, lying, etc.) [25, 40, 24, 41, 42, 23]. Accelerometers are inexpensive, require relatively low power [43], and are embedded in most of today’s cellular phones. Several papers have reported high recognition accuracy 92% [25], 95% [44], 97% [41], and up to 98% [45], under different evaluation methodologies. However, other daily activities such as eating, working at a computer, or brushing teeth, are confusing from the acceleration point of view. For instance, eating might be confused with brushing teeth due to arm motion [22]. The impact of the sensor specifications have also been analyzed. In fact, Maurer et al. [22] studied the behavior of the recognition accuracy as a function of the accelerometer sampling rate (which lies between 10 Hz [4] and 100 Hz [21]). Interestingly, they found that no significant gain in accuracy is achieved above 20 Hz for ambulation activities. In addition, the amplitude of the accelerometers varies from $\pm 2g$ [22], up to $\pm 6g$ [45], yet $\pm 2g$ was shown to be sufficient to recognize ambulation activities [22]. The placement of the accelerometer is another important point of discussion: He et al. [25] found that the best place to wear the accelerometer is inside the trousers pocket. Instead, other studies suggest that the accelerometer should be placed in a bag carried by the user [22], on the belt [46], or on the dominant wrist [20]. At the end, the optimal position where to place the accelerometer depends on the application and the type of activities to be recognized.

### 2.1.2.3 Location

The Global Positioning System (GPS) enables all sort of location based services. Current cellular phones are equipped with GPS devices, making this sensor very convenient for context-aware applications, including the recognition of the user’s transportation mode [43]. The place where the user is can also be helpful to infer their activity using ontological
reasoning [27]. As an example, if a person is at a park, they are probably not brushing their teeth but might be running or walking. And, information about places can be easily obtained by means of the Google Places Web Service [47], among other tools. However, GPS devices do not work well indoors and they are relatively expensive in terms of energy consumption, especially for real-time tracking applications [43]. Consequently, this sensor is usually employed along with accelerometers [27]. Finally, location data have privacy issues because users are not always willing to be tracked. Encryption, obfuscation, and anonymization are some of the techniques available to ensure privacy in location data [48, 49, 50].

2.1.2.4 Physiological Signals

Vital signs data (e.g., heart rate, respiration rate, skin temperature, skin conductivity, ECG, etc.) have also been considered in a few works [4]. Tapia et al. [20] proposed an activity recognition system that combines data from five triaxial accelerometers and a heart rate monitor. However, they concluded that the heart rate is not useful in a HAR context because after performing physically demanding activities (e.g., running) the heart rate remains at a high level for a while, even if the individual is lying or sitting. In this dissertation, it was shown that, by means of structural feature extraction, vital signs can be exploited to improve recognition accuracy. Now, in order to measure physiological signals, additional sensors would be required, thereby increasing the system cost and increasing the level of obtrusiveness [16].

2.1.3 Obtrusiveness

To be successful in practice, HAR systems should not require the user to wear many sensors nor interact too often with the application. Nevertheless, the more sources of data available, the richer the information that can be extracted from the measured attributes. There are systems which require the user to wear four or more accelerometers [23, 51, 20], or carry a heavy rucksack with recording devices [16]. These configurations may be
uncomfortable, invasive, expensive, and hence not suitable for activity recognition. Other systems are able to work with rather unobtrusive hardware. For instance, a sensing platform that can be worn as a sport watch is presented in [22]. Finally, the systems introduced in [43, 29] recognize activities with a cellular phone only.

2.1.4 Data Collection Protocol

The procedure followed by the individuals while collecting data is critical in any HAR. In 1999, Foerster et al. [38] demonstrated 95.6% of accuracy for ambulation activities in a controlled data collection experiment, but in a naturalistic environment (i.e., outside of the laboratory), the accuracy dropped to 66%! The number of individuals and their physical characteristics are also crucial factors in any HAR study. A comprehensive study should consider a large number of individuals with diverse characteristics in terms of gender, age, height, weight, and health conditions. This is with the purpose of ensuring flexibility to support new users without the need of collecting additional training data.

2.1.5 Recognition Performance

The performance of a HAR system depends on several aspects, such as (1) the activity set (2) the quality of the training data, (3) the feature extraction method, and (4) the learning algorithm. In the first place, each set of activities brings a completely different pattern recognition problem. For example, discriminating among *walking*, *running*, and *standing still* [52], turns out to be much easier than incorporating more complex activities such as *watching TV*, *eating*, *walking upstairs*, and *walking downstairs* [23]. Secondly, there should be a sufficient amount of training data, similar to the expected testing data. Finally, a comparative evaluation of several learning methods is desirable as each dataset exhibits distinct characteristics that can be either beneficial or detrimental for a particular method. Such interrelationship among datasets and learning methods can be very hard to analyze theoretically, which accentuates the need of an experimental study. In order to quantitatively understand the recognition performance, some standard metrics are used,
e.g., accuracy, recall, precision, F-measure, Kappa statistic, etc. These metrics will be discussed in Section 2.2.

2.1.6 Energy Consumption

Context-aware applications rely on mobile devices—such as sensors and cellular phones—which are generally energy constrained. In most scenarios, extending the battery life is a desirable feature, especially for medical and military applications that are compelled to deliver critical information. Surprisingly, most HAR schemes do not formally analyze energy expenditures, which are mainly due to processing, communication, and visualization tasks. Communication is often the most expensive operation, so the designer should minimize the amount of transmitted data. In most cases, short range wireless networks (e.g., Bluetooth or Wi-Fi) should be preferred over long range networks (e.g., cellular network or WiMAX) as the former require lower power. Some typical energy saving mechanisms are data aggregation and compression yet they involve additional computations that may affect the application performance. Another approach is to carry out feature extraction and classification in the integration device, so that raw signals would not have to be continuously sent to the server [34, 27]. This will be discussed in Section 2.1.7. Finally, since all sensors may not be necessary simultaneously, turning some of them off or reducing their sampling/transmission rate is very convenient to save energy. For example, if the user’s activity is sitting or standing still, the GPS sensor may be turned off [43].

2.1.7 Processing

Another important point of discussion is where the recognition task should be done, whether in the server or in the integration device. On one hand, a server is expected to have huge processing, storage, and energy capabilities, allowing to incorporate more complex methods and models. On the other hand, a HAR system running on a mobile device should substantially reduce energy expenditures, as raw data would not have to be continuously sent to a server for processing. The system would also become more robust and
responsive because it would not depend on unreliable wireless communication links, which may be unavailable or error prone; this is particularly important for medical or military applications that require real-time decision making. Finally, a mobile HAR system would be more scalable since the server load would be alleviated by the locally performed feature extraction and classification computations. However, implementing activity recognition in mobile devices becomes challenging because they are still constrained in terms of processing, storage, and energy. Hence, feature extraction and learning methods should be carefully chosen to guarantee a reasonable response time and battery life. For instance, classification algorithms such as Instance Based Learning [28] and Bagging [53] are very expensive in their evaluation phase, which makes them not convenient for HAR.

2.1.8 User Flexibility

There is an open debate on the design of any activity recognition model. Some authors claim that, as people perform activities in a different manner (due to age, gender, weight, and so on), a specific recognition model should be built for each individual [26]. This implies that the system should be re-trained for each new user. Other studies rather emphasize the need of a monolithic recognition model, flexible enough to work with different users. Consequently, two types of analyses have been proposed to evaluate activity recognition systems: subject-dependent and subject-independent evaluations [20]. In the first one, a classifier is trained and tested for each individual with his/her own data and the average accuracy for all subjects is computed. In the second one, only one classifier is built for all individuals and the evaluation is carried out by cross validation or leave-one-individual-out analysis. It is worth to highlight that, in some cases, it would not be convenient to train the system for each new user, especially when (1) there are too many activities; (2) some activities are not desirable for the subject to carry out (e.g., falling downstairs); or (3) the subject would not cooperate with the data collection process (e.g., patients with dementia and other mental pathologies). On the other hand, an elderly lady would surely walk quite differently than a ten-year-old boy, thereby challenging a single model to recognize activities
regardless of the subject’s characteristics. A solution to the dichotomy of the monolithic vs. particular recognition model is to create groups of users with similar characteristics. Additional design considerations related to this matter will be discussed in Section 6.2.

2.2 Activity Recognition Methods

Section 1.2 showed that, to enable the recognition of human activities, raw data have to first pass through the process of feature extraction. Then, the recognition model is built from the set of feature instances by means of machine learning techniques. Once the model is trained, unseen instances (i.e., time windows) can be evaluated in the recognition model, yielding a prediction on the performed activity. Next, the most noticeable approaches in feature extraction and learning will be covered.

2.2.1 Feature Extraction

Human activities are performed during relatively long periods of time (in the order of seconds or minutes) compared to the sensors’ sampling rate (which can be up to 250 Hz). Besides, a single sample on a specific time instant (e.g., the Y-axis acceleration is 2.5g, or the heart rate is 130 bpm) does not provide sufficient information to describe the performed activity. Thus, activities need to be recognized in a time window basis rather than in a sample basis. Now, the question is: how can two given time windows be compared? It would be nearly impossible for the signals to be exactly identical, even if they come from the same subject performing the same activity. This is the main motivation for applying feature extraction (FE) methodologies to each time window: filtering relevant information and obtaining quantitative measures that allow signals to be compared.

In general, two approaches have been proposed to extract features from time series data: statistical and structural [54]. Statistical methods, such as the Fourier transform and the Wavelet transform, use quantitative characteristics of the data to extract features, whereas structural approaches take into account the morphological interrelationship among
data. The criterion to choose either of these methods is certainly subject to the nature of the given signal.

Figure 2.1 displays the process to transform the raw time series dataset —which can be from acceleration, environmental variables, or vital signs— into a set of feature vectors. $w$ is the window consecutive; $s_i$ is the sampling rate for the group of sensors $i$, where sensors in the same group have the same sampling rate; and $f_i$ is each of the extracted features. Each instance in the processed dataset corresponds to the set of features computed from an entire window in the raw dataset.
The next sections will cover the most common FE techniques for each of the measured attributes, i.e., acceleration, environmental signals, and vital signs. GPS data are not considered in this section since they are mostly used to compute the speed [16, 43] or include some knowledge about the place where the activity is being performed [27].

### 2.2.1.1 Acceleration

Acceleration signals (see Figure 2.2) are highly fluctuating and oscillatory, which makes it difficult to recognize the underlying patterns using their raw values. Existing HAR systems based on accelerometer data employ statistical feature extraction and, in most of the cases, either time- or frequency-domain features. Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA) have also been applied with promising results [41], as well as autoregressive model coefficients [25]. All these techniques are conceived to handle the high variability inherent to acceleration signals. Table 2.2 summarizes the feature extraction methods for acceleration signals. The definition of some of the most widely used features [42] are listed below for a given signal $Y = \{y_1, ..., y_n\}$.

- Central tendency measures such as the arithmetic mean $\bar{y}$ and the root mean square (RMS) (Equations 2.1 and 2.2).

- Dispersion metrics such as the standard deviation $\sigma_y$, the variance $\sigma_y^2$, and the mean absolute deviation (MAD) (Equations 2.3, 2.4, and 2.5).

- Domain transform measures such as the energy, where $F_i$ is the $i$-th component of the Fourier Transform of $Y$ (Equation 2.6).
Table 2.2: Summary of feature extraction methods for acceleration signals.

<table>
<thead>
<tr>
<th>Group</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time domain</td>
<td>Mean, standard deviation, variance, interquartile range (IQR), mean absolute deviation (MAD), correlation between axes, entropy, and kurtosis [42, 23, 22, 16, 20, 51, 21].</td>
</tr>
<tr>
<td>Frequency domain</td>
<td>Fourier Transform (FT) [23, 42] and Discrete Cosine Transform (DCT) [55].</td>
</tr>
<tr>
<td>Others</td>
<td>Principal Component Analysis (PCA) [41, 55], Linear Discriminant Analysis (LDA) [42], Autoregresive Model (AR), and HAAR filters [24].</td>
</tr>
</tbody>
</table>

\[
\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \quad (2.1)
\]

\[
RMS(Y) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} y_i^2} \quad (2.2)
\]

\[
\sigma_y = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2} \quad (2.3)
\]

\[
\sigma_y^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2 \quad (2.4)
\]

\[
MAD(Y) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} |y_i - \bar{y}|} \quad (2.5)
\]

\[
Energy(Y) = \frac{\sum_{i=1}^{n} F_i^2}{n} \quad (2.6)
\]

2.2.1.2 Environmental Variables

Environmental attributes, along with acceleration signals, have been used to enrich context awareness. For instance, the values from air pressure and light intensity are helpful to determine whether the individual is outdoors or indoors [57]. Also, audio signals are
Table 2.3: Summary of feature extraction methods for environmental variables.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>Time-domain [16]</td>
</tr>
<tr>
<td>Audio</td>
<td>Speech recognizer [56]</td>
</tr>
<tr>
<td>Barometric pressure</td>
<td>Time-domain and frequency-domain [57]</td>
</tr>
<tr>
<td>Humidity</td>
<td>Time-domain [16]</td>
</tr>
<tr>
<td>Light</td>
<td>Time-domain [58] and frequency-domain [16]</td>
</tr>
<tr>
<td>Temperature</td>
<td>Time-domain [16]</td>
</tr>
</tbody>
</table>

useful to conclude that the user is having a conversation rather than listening to music [16]. Table 2.3 summarizes the feature extraction methods for environmental attributes.

2.2.1.3 Physiological Signals

Since the incorporation of physiological signals is one of the main contributions of this dissertation, a detailed description of feature extraction from vital signs is provided in Chapter 3.

2.2.1.4 Selection of the Window Length

In accordance to Definition 2, dividing the measured time series in time windows is a convenient solution to relax the HAR problem. Therefore, a key factor is the selection of the window length because the computational complexity of any FE method depends on the number of samples. Having rather short windows may enhance the FE performance, but would entail higher overhead due to the recognition algorithm being triggered more frequently. Besides, short time windows may not provide sufficient information to fully describe the performed activity. Conversely, if the windows are too long, there might be more than one activity within a single time window [43]. Different window lengths have been used in the literature: 0.08s [26], 1s [43], 1.5s [39], 3s [59], 5s [55], 7s [60], 12s [33], or up to 30s [20]. Of course, this decision is conditioned to the activities to be recognized and the measured attributes. The heart rate signal, for instance, required 30s time windows in [20]. Instead, for activities such as swallowing, 1.5s time windows were employed.
Time windows can also be either overlapping [23, 20, 61] or disjoint [43], [39], [59], [55]. Overlapping time windows are intended to handle transitions more accurately, although using small non-overlapping time windows, misclassifications due to transitions are negligible.

2.2.1.5 Feature Selection

Some features in the processed dataset might contain redundant or irrelevant information that can negatively affect the recognition accuracy. Then, implementing techniques for selecting the most appropriate features is a suggested practice to reduce computations and simplify the learning models. The Bayesian Information Criterion (BIC) and the Minimum Description Length (MDL) have been widely used for general machine learning problems. In HAR, a common method is the Minimum Redundancy and Maximum Relevance (MRMR) [62], utilized in [17]. In that work, the minimum mutual information between features is used as criteria for minimum redundancy and the maximal mutual information between the classes and features is used as criteria for maximum relevance. In contrast, Maurer et al. [58] applied a Correlation-based Feature Selection (CFS) approach [63], taking advantage of the fact that this method is built in WEKA [30]. CFS works under the assumption that features should be highly correlated with the given class but uncorrelated with each other. Iterative approaches have also been evaluated to select features. Since the number of feature subsets is $O(2^n)$, for $n$ features, evaluating all possible subsets is not computationally feasible. Hence, metaheuristic methods such as multiobjective evolutionary algorithms have been employed to explore the space of possible feature subsets [64].

2.2.2 Learning

In a machine learning context, patterns are to be discovered from a set of given examples or observations denominated instances. Such input set is called training set. In our specific case, each instance is a feature vector extracted from signals within a time window. The examples in the training set may or may not be labeled, i.e., associated to a known class (e.g., walking, running, etc.). In some cases, labeling data is not feasible because it may
require an expert to manually examine the examples and assign a label based upon their experience. This process is usually tedious, expensive, and time consuming in many data mining applications.

There exist two learning approaches, namely *supervised* and *unsupervised* learning, which deal with labeled and unlabeled data, respectively. Since a human activity recognition system should return a label such as *walking, sitting, running*, etc., most HAR systems work in a supervised fashion. Indeed, it might be very hard to discriminate activities in a completely unsupervised context. Some other systems work in a semi-supervised fashion allowing part of the data to be unlabeled.

### 2.2.2.1 Supervised Learning

Labeling sensed data from individuals performing different activities is a relatively easy task. Some systems [20, 58] store sensor data in a non-volatile medium while a person from the research team supervises the collection process and manually registers activity labels and time stamps. Other systems feature a mobile application that allows the user to select the activity to be performed from a list [33]. In this way, each sample is matched to an activity label, and then stored in the server.

Supervised learning — referred to as *classification* for discrete-class problems — has been a very productive field, bringing about a great number of algorithms. Table 2.4 summarizes the most important classifiers in Human Activity Recognition and their description is included below.

- **Decision trees** build a hierarchical model in which attributes are mapped to nodes and edges represent the possible attribute values. Each branch from the root to a leaf node is a classification rule. C4.5 is perhaps the most widely used decision tree classifier and is based on the concept of information gain to select which attributes should be placed in the top nodes [69]. Decision trees can be evaluated in $O(\log n)$ for $n$ attributes, and usually generate models that are easy to understand by humans.
Table 2.4: Classification algorithms used by state-of-the-art human activity recognition systems.

<table>
<thead>
<tr>
<th>Type</th>
<th>Classifiers</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision tree</td>
<td>C4.5, ID3</td>
<td>[23, 51, 17, 22]</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Naïve Bayes and Bayesian Networks</td>
<td>[20, 23, 17, 34]</td>
</tr>
<tr>
<td>Instance Based</td>
<td>k-nearest neighbors</td>
<td>[22, 17]</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>Multilayer Perceptron</td>
<td>[65]</td>
</tr>
<tr>
<td>Domain transform</td>
<td>Support Vector Machines</td>
<td>[41, 40, 25]</td>
</tr>
<tr>
<td>Fuzzy Logic</td>
<td>Fuzzy Basis Function, Fuzzy Inference System</td>
<td>[21, 42, 66]</td>
</tr>
<tr>
<td>Regression methods</td>
<td>MLR, ALR</td>
<td>[27, 33]</td>
</tr>
<tr>
<td>Markov models</td>
<td>Hidden Markov Models, Conditional Random Fields</td>
<td>[67, 68]</td>
</tr>
<tr>
<td>Classifier ensembles</td>
<td>Boosting and Bagging</td>
<td>[59, 33]</td>
</tr>
</tbody>
</table>

- **Bayesian** methods calculate posterior probabilities for each class using estimated conditional probabilities from the training set. The **Bayesian Network** (BN) [70] classifier and **Naïve Bayes** (NB) [71] —which is a specific case of BN— are the principal exponents of this family of classifiers. A key issue in Bayesian Networks is the topology construction, as it is necessary to make assumptions on the independence among features. For instance, the NB classifier assumes that all features are conditionally independent given a class value, yet such assumption does not hold in many cases. As a matter of fact, acceleration signals are highly correlated, as well as physiological signals such as heart rate, respiration rate, and ECG amplitude.

- **Instance based learning** (IBL) [53] methods classify an instance based upon the most similar instance(s) in the training set. For that purpose, they define a distance function to measure similarity between each pair of instances. This makes IBL classifiers quite expensive in their evaluation phase as each new instance to be classified needs to be compared to the entire training set. Such high cost in terms of computation and storage, makes IBL models not convenient to be implemented in a mobile device.

- **Support Vector Machines** (SVM) [72] and **Artificial Neural Networks** (ANN) [73] have also been broadly used in HAR although they do not provide a set of rules under-
standable to humans. Instead, knowledge is hidden within the model, which may hinder the analysis and incorporation of additional reasoning. SVMs rely on kernel functions that project all instances to a higher dimensional space with the aim of finding a linear decision boundary (i.e., a hyperplane) to partition the data. Neural networks replicate the behavior of biological neurons in the human brain, propagating activation signals and encoding knowledge in the network links. Besides, ANNs have been shown to be universal function approximators. The high computational cost and the need for large amount of training data are two common drawbacks of neural networks.

- **Classifier ensembles** combine the output of several classifiers to improve classification accuracy. Some examples are *bagging*, *boosting*, and *stacking* [53]. Classifier ensembles are clearly more expensive, computationally speaking, as they require several models to be trained and evaluated. Section 2.4 and Chapter 5 elaborate on these methods.

### 2.2.2.2 Semi-supervised Learning

Relatively few approaches have implemented activity recognition in a semi-supervised fashion, thus, having part of the data without labels [74, 75, 76, 77, 78]. In practice, annotating data might be difficult in some scenarios, particularly when the granularity of the activities is very high or the user is not willing to cooperate with the collection process. Since semi-supervised learning is a minority in HAR, there are no standard algorithms or methods, but each system implements its own approach. Section 2.3.3 provides more details on the state-of-the-art semi-supervised activity recognition approaches.

### 2.2.2.3 Evaluation Methodologies

When evaluating a machine learning algorithm, the training and testing datasets should be disjoint. This is with the aim of assessing how effective the algorithm is to model unseen data. A very intuitive approach is called *random split* and it simply divides the entire dataset in two partitions: one for training and the other one for testing —usually, two
thirds of the data are for training and the remaining one third is for testing. However, a random split is highly biased by the dataset partition. If the training set is not diverse enough, the evaluation metrics would not reflect the actual performance of the classifier. Therefore, a more robust approach is the cross validation. In a $k$-fold cross validation, the dataset is divided in $k$ equally-sized folds. In the first iteration, the very first fold is used as testing set while the remaining $k - 1$ folds constitute the training set. The process is repeated $k$ times, using each fold as a testing set and the remaining ones for training. In the end, the evaluation metrics (e.g., accuracy, precision, recall, etc.) are averaged out over all iterations. In practice, a 10-fold cross validation is the most widely accepted methodology to calculate the accuracy of a certain classifier (see Figure 2.3). Yet, if the goal is to compare two classifiers in order to choose the most accurate one, a $5 \times 2$-fold cross validation with a paired $t$-test is recommended [79]. This is nothing but repeating a 2-fold cross validation five times with different dataset partitions, which is often achieved by using different seeds for the random number generator. The result of a $5 \times 2$-fold cross validation is shown Table 2.5, where the accuracies $a_i$ and $b_i$ are for classifiers $A$ and $B$, respectively, throughout the five repetitions. The next step is to apply a statistical paired $t$-test to find the most accurate classifier if there is significant statistical difference among their accuracies. The null hypothesis is that both classifiers $A$ and $B$ have the same accuracy—or conversely, the same error rate. The $\tilde{t}$ statistic is defined in Equation 2.7 as follows:

$$
\tilde{t} = \frac{p_i^{(1)} - p_i^{(2)}}{\sqrt{\frac{1}{5} \sum_{i=1}^{5} s_i^2}}
$$

(2.7)

where $p_i^{(j)}$ is the difference between the accuracies of both classifiers in the $j$-th iteration for $1 \leq j \leq 2$ and the $i$-th replication; $s_i^2 = (p_i^{(1)} - \bar{p})^2 + (p_i^{(2)} - \bar{p})^2$ is the estimated variance from the $i$-th replication for $1 \leq i \leq 5$; and $\bar{p}$ is the average of $p_i^{(1)}$ and $p_i^{(2)}$. Under the null hypothesis, $\tilde{t}$ follows the Student’s $t$ distribution with five degrees of freedom. It has been shown that the $5 \times 2$-fold cross validation with the paired $t$-test is more powerful than the
non-parametric McNemar’s test and provides a better measure of the variations due to the choice of the training set [79].

### 2.2.2.4 Evaluation Metrics in Machine Learning

In general, the selection of a classification algorithm for HAR has been merely supported by empirical evidence. The vast majority of the studies use cross validation with statistical tests to compare classifiers’ performance for a particular dataset. The classification results for a particular method can be organized in a confusion matrix $M_{n \times n}$ for a classification problem with $n$ classes. This is a matrix such that the element $M_{ij}$ is the number of instances from class $i$ that were classified as class $j$. The following values can be obtained from the confusion matrix in a binary classification problem:

- **True Positives (TP):** The number of positive instances that were classified as positive.
• **True Negatives** (TN): The number of negative instances that were classified as negative.

• **False Positives** (FP): The number of negative instances that were classified as positive.

• **False Negatives** (FN): The number of positive instances that were classified as negative.

The **accuracy** is the most standard metric to summarize the overall classification performance for all classes and it is defined as follows:

\[
    \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}
\]

The **precision** — often referred to as *positive predictive value* — is the ratio of correctly classified positive instances to the total number of instances classified as positive:

\[
    \text{Precision} = \frac{TP}{TP + FP} \tag{2.9}
\]

The **recall**, also called *true positive rate*, is the ratio of correctly classified positive instances to the total number of positive instances:

\[
    \text{Recall} = \frac{TP}{TP + FN} \tag{2.10}
\]

The *F-measure* combines precision and recall in a single value:

\[
    F - \text{measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.11}
\]

Finally, the **false positive rate** (FPR) and the **false negative rate** (FNR) are defined as follows:

\[
    \text{FPR} = \frac{FP}{TN + FP} \tag{2.12}
\]
\[
FNR = \frac{FN}{TP + FN} \quad (2.13)
\]

Although defined for binary classification, these metrics can be generalized for a problem with \( n \) classes. In such case, an instance could be positive or negative according to a particular class, e.g., positives might be all instances of \textit{running} while negatives would be all instances other than \textit{running}.

### 2.2.2.5 Machine Learning Tools

The Waikato Environment for Knowledge Analysis (WEKA) [30] is certainly the best known tool in the machine learning research community. It contains implementations of a number of learning algorithms and it allows to easily evaluate them for a particular dataset using cross validation and random split, among others. WEKA also offers a Java API that facilitates the incorporation of new learning algorithms and evaluation methodologies on top of the pre-existing framework. One of the limitations of WEKA [30] and other machine learning platforms such as the Java Data Mining (JDM) platform [31] is that they are not optimized to work on current mobile platforms. In that direction, this dissertation introduces MECLA [34], a mobile platform for the evaluation of classification algorithms under the Android platform.

### 2.3 Evaluation of HAR Systems

In this dissertation, a new two-level taxonomy is proposed to categorize HAR systems (see Figure 2.4). The first level relates to the learning approach, which can be either supervised or semi-supervised. In the second level, according to the response time, supervised approaches can work either online or offline. The former provide immediate feedback on the performed activities. The latter either need more time to recognize activities due to high computational demands, or are intended for applications that do not require real-time feedback. To the best of our knowledge, current semi-supervised systems have been im-
implemented and evaluated offline. This taxonomy has been adopted as the systems within each class have very different purposes and associated challenges and should be evaluated separately. For instance, a very accurate fully supervised approach might not work well in a semi-supervised scenario, whereas an effective offline system may not be able to run online due to processing constraints. Furthermore, we found a significant number of systems that fall in each group, which also favors the comparison and analysis by means of the proposed taxonomy.

The qualitative evaluation encompasses the following aspects:

- Recognized activities (Table 2.1)
- Type of sensors and the measured attributes (Section 2.1.2)
- Integration device
- Level of obtrusiveness, which could be low, medium, or high.
- Type of data collection protocol, which could be either a controlled or a naturalistic experiment.
- Level of energy consumption, which could be low, medium, or high.
• User flexibility level, which could be either user-specific or monolithic.

• Feature extraction method(s)

• Learning algorithm(s)

• Overall accuracy for all activities

The abbreviations and acronyms are defined in Table 2.6. The most important works on online HAR are described next.

2.3.1 Online HAR systems

Applications of online activity recognition systems can be easily visualized. In healthcare, continuously monitoring patients with physical or mental pathologies becomes crucial for their protection, safety, and recovery. Likewise, interactive games or simulators may enhance user’s experience by considering activities and gestures. Table 2.7 summarizes the online state-of-the-art activity recognition approaches.

2.3.1.1 eWatch

Maurer et al. [22] introduced eWatch as an online activity recognition system which embeds sensors and a microcontroller within a device that can be worn as a sport watch. Four sensors are included, namely an accelerometer, a light sensor, a thermometer, and a microphone. These are passive sensors and, as they are embedded in the device, no wireless communication is needed; thus, eWatch is very energy efficient. Using a C4.5 decision tree and time-domain feature extraction, the overall accuracy was up to 92.5% for six ambulation activities, although they achieved less than 70% for activities such as descending and ascending. The execution time for feature extraction and classification is less than 0.3 ms, which makes the system very responsive. However, in eWatch, data were collected under controlled conditions, i.e., a lead experimenter supervised and gave specific guidelines to the subjects on how to perform the activities [22]. Section 2.1.4 describes the disadvantages of this approach.
Table 2.6: List of abbreviations and acronyms.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DD</td>
<td>3D Deviation of the acceleration signals</td>
</tr>
<tr>
<td>ACC</td>
<td>Accelerometers</td>
</tr>
<tr>
<td>AMB</td>
<td>Ambulation activities (see Table 2.1)</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>ALR</td>
<td>Additive Logistic Regression classifier</td>
</tr>
<tr>
<td>AR</td>
<td>Autoregressive Model Coefficients</td>
</tr>
<tr>
<td>AV</td>
<td>Angular velocity</td>
</tr>
<tr>
<td>BN</td>
<td>Bayesian Network Classifier</td>
</tr>
<tr>
<td>CART</td>
<td>Classification And Regression Tree</td>
</tr>
<tr>
<td>DA</td>
<td>Daily activities (see Table 2.1)</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DT</td>
<td>Decision Tree-Based Classifier</td>
</tr>
<tr>
<td>DTW</td>
<td>Dynamic Time Warping</td>
</tr>
<tr>
<td>ENV</td>
<td>Environmental sensors</td>
</tr>
<tr>
<td>FBF</td>
<td>Fuzzy Basis Function</td>
</tr>
<tr>
<td>FD</td>
<td>Frequency-domain features</td>
</tr>
<tr>
<td>GYR</td>
<td>Gyroscope</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Models</td>
</tr>
<tr>
<td>HRB</td>
<td>Heart Rate Beats above the resting heart rate</td>
</tr>
<tr>
<td>HRM</td>
<td>Heart Rate Monitor</td>
</tr>
<tr>
<td>HW</td>
<td>Housework activities (see Table 2.1)</td>
</tr>
<tr>
<td>KNN</td>
<td>k-Nearest Neighbors classifier</td>
</tr>
<tr>
<td>LAB</td>
<td>Laboratory controlled experiment</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares algorithm</td>
</tr>
<tr>
<td>MIL</td>
<td>Military activities</td>
</tr>
<tr>
<td>MNL</td>
<td>Monolithic classifier (subject independent)</td>
</tr>
<tr>
<td>NAT</td>
<td>Naturalistic experiment</td>
</tr>
<tr>
<td>NB</td>
<td>The Naïve Bayes classifier</td>
</tr>
<tr>
<td>NDDF</td>
<td>Normal Density Discriminant Function</td>
</tr>
<tr>
<td>N/S</td>
<td>Not Specified</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>PHO</td>
<td>Activities related to phone usage (see Table 2.1)</td>
</tr>
<tr>
<td>PR</td>
<td>Polynomial Regression</td>
</tr>
<tr>
<td>RFIS</td>
<td>Recurrent Fuzzy Inference System</td>
</tr>
<tr>
<td>SD</td>
<td>Subject Dependent evaluation</td>
</tr>
<tr>
<td>SFFS</td>
<td>Sequential Forward Feature Selection</td>
</tr>
<tr>
<td>SI</td>
<td>Subject Independent evaluation</td>
</tr>
<tr>
<td>SMA</td>
<td>Signal Magnitude Area</td>
</tr>
<tr>
<td>SMCRF</td>
<td>Semi-Markovian Conditional Random Field</td>
</tr>
<tr>
<td>SPC</td>
<td>User-specific classifier (subject dependent)</td>
</tr>
<tr>
<td>SPI</td>
<td>Spiroergometry</td>
</tr>
<tr>
<td>TA</td>
<td>Tilt Angle</td>
</tr>
<tr>
<td>TD</td>
<td>Time-domain features</td>
</tr>
<tr>
<td>TF</td>
<td>Transient Features [33]</td>
</tr>
<tr>
<td>TR</td>
<td>Transitions between activities</td>
</tr>
<tr>
<td>UB</td>
<td>Upper body activities (see Table 2.1)</td>
</tr>
<tr>
<td>VS</td>
<td>Vital sign sensors</td>
</tr>
</tbody>
</table>
Table 2.7: Summary of state-of-the-art in online human activity recognition systems.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Activities</th>
<th>Sensors</th>
<th>ID</th>
<th>Obtrusive</th>
<th>Experiment</th>
<th>Energy</th>
<th>Flexibility</th>
<th>Processing</th>
<th>Feature</th>
<th>Learning</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ermes [51]</td>
<td>AMB (5)</td>
<td>ACC (wrist, ankle, chest)</td>
<td>PDA</td>
<td>High</td>
<td>N/S</td>
<td>High</td>
<td>SPC</td>
<td>High</td>
<td>TD, FD</td>
<td>DT</td>
<td>94%</td>
</tr>
<tr>
<td>eWatch [22]</td>
<td>AMB (6)</td>
<td>ACC, ENV (wrist)</td>
<td>Custom</td>
<td>Low</td>
<td>LAB</td>
<td>Low</td>
<td>MNL</td>
<td>Low</td>
<td>TD, FD</td>
<td>C4.5, NB</td>
<td>94%</td>
</tr>
<tr>
<td>Tapia [20]</td>
<td>EXR (30)</td>
<td>ACC (5 places), HRM</td>
<td>Laptop</td>
<td>High</td>
<td>LAB</td>
<td>High</td>
<td>Both</td>
<td>High</td>
<td>TD, FD, HB</td>
<td>C4.5, NB</td>
<td>86% (SD), 56% (SI)</td>
</tr>
<tr>
<td>Vigilante [34]</td>
<td>AMB (5)</td>
<td>ACC and VS (chest)</td>
<td>Phone</td>
<td>Medium</td>
<td>NAT</td>
<td>Medium</td>
<td>MNL</td>
<td>Low</td>
<td>TD, FD, TF</td>
<td>C4.5</td>
<td>96.8% (SD), 92.6% (SI)</td>
</tr>
<tr>
<td>Kao [21]</td>
<td>AMB, DA (7)</td>
<td>ACC (wrist)</td>
<td>Custom</td>
<td>Low</td>
<td>N/S</td>
<td>Medium</td>
<td>MNL</td>
<td>Low</td>
<td>TD, LDA</td>
<td>FBF</td>
<td>94.71%</td>
</tr>
<tr>
<td>Brezmes [28]</td>
<td>AMB (5)</td>
<td>ACC (phone)</td>
<td>Phone</td>
<td>Low</td>
<td>N/S</td>
<td>Low</td>
<td>SPC</td>
<td>High</td>
<td>TD, FD</td>
<td>KNN</td>
<td>80%</td>
</tr>
<tr>
<td>COSAR [27]</td>
<td>AMB, DA (10)</td>
<td>ACC (watch, phone), GPS</td>
<td>Phone</td>
<td>Low</td>
<td>NAT</td>
<td>Medium</td>
<td>MNL</td>
<td>Medi</td>
<td>TD</td>
<td>COSAR</td>
<td>93%</td>
</tr>
<tr>
<td>ActiServ [29, 26]</td>
<td>AMB, PHO (11)</td>
<td>ACC (phone)</td>
<td>Phone</td>
<td>Low</td>
<td>N/S</td>
<td>Low</td>
<td>SPC</td>
<td>High $\bar{y}, \sigma^2_y$</td>
<td>RFIS</td>
<td>71% - 98%</td>
<td></td>
</tr>
</tbody>
</table>
2.3.1.2 Vigilante

Vigilante [34] is proposed in this dissertation (see Chapter 4), as a mobile application for real-time human activity recognition under the Android platform. The Zephyr’s BioHarness BT [80] chest sensor strap was used to measure acceleration and physiological signals such as heart rate, respiration rate, breath waveform amplitude, and skin temperature, among others. Statistical time- and frequency-domain features were extracted from acceleration signals while transient features and linear regression were applied to physiological signals. The C4.5 and the Additive Logistic Regression classifiers were employed to recognize five activities with an overall accuracy of up to 96.8%. The application can run for up to 12.5 continuous hours with a response time of no more than 8% of the window length. Unlike other approaches, Vigilante was evaluated completely online to provide more realistic results. Vigilante is moderately energy efficient because it requires permanent Bluetooth communication between the sensor strap and the phone.

2.3.1.3 Tapia

This system [20] recognizes 17 ambulation and gymnasium activities such as lifting weights, rowing, doing push ups, etc., with different intensities (a total of 30 activities). A comprehensive study was carried out, including 21 participants and both subject-dependent and subject-independent studies. The average classification accuracy was reported to be 94.6% for subject-dependent analysis whereas a 56% of accuracy was reached in the subject-independent evaluation. If intensities are not considered, the overall subject-independent accuracy is 80.6%. This system works with very obtrusive hardware, i.e., five accelerometers were placed on the user’s dominant arm and wrist, hip, thigh, and ankle, as well as a heart rate monitor on the chest. Besides, all these sensors require wireless communication, involving high energy consumption. Finally, the integration device is a laptop, which allows for better processing capabilities, but prevents portability and pervasiveness.
2.3.1.4 ActiServ

In 2010, Berchtold et al. introduced ActiServ as an activity recognition service for mobile phones [29, 26]. The system was implemented on the Neo FreeRunner phone. They make use of a fuzzy inference system to classify ambulation and phone activities based on the signals given by the phone’s accelerometer only. This makes ActiServ a very energy efficient and portable system. The overall accuracy varies between 71% and 97%. However, in order to reach the top accuracy level, the system requires a runtime duration in the order of days! When the algorithms are executed to meet a real-time response time, the accuracy drops to 71%. ActiServ can also reach up to 90% after personalization, in other words, a subject-dependent analysis. From the reported confusion matrices, the activity labeled as walking was often confused with cycling, while standing and sitting could not be differentiated when the cellphone’s orientation was changed.

2.3.1.5 COSAR

Riboni et al. [27] presented COSAR, a framework for context-aware activity recognition using statistical and ontological reasoning under the Android platform. The system recognizes ambulation activities as well as brushing teeth, strolling, and writing on a blackboard. COSAR gathers data from two accelerometers, one in the phone and another on the individual’s wrist, as well as from the cellphone’s GPS. Since COSAR makes use of the GPS sensor, it was catalogued as a moderately energy efficient system. COSAR uses an interesting concept of potential activity matrix to filter activities based upon the user’s location. For instance, if the individual is in the kitchen, he or she is probably not cycling. Another contribution is the statistical classification of activities with a historical variant. For example, if the predictions for the last five time windows were \{jogging, jogging, walking, jogging, jogging\}, the third window was likely a misclassification (e.g., due to the user performing some atypical movement) and the algorithm should automatically correct it. However, this introduces an additional delay to the system’s response, according to the number of windows analyzed for the historical variant. The overall accuracy was roughly
93% though, in some cases, standing still was confused with writing on a blackboard, as well as hiking up with hiking down.

2.3.1.6 Kao

Kao et al. [21] presented a portable device for online activity detection. A triaxial accelerometer is placed on the user’s dominant wrist, sampling at 100 Hz. They apply time domain features and the Linear Discriminant Analysis (LDA) to reduce the dimension of the feature space. Then, a Fuzzy Basis Function learner —which uses fuzzy If-Then rules— classifies the activities. An overall accuracy of 94.71% was reached for seven activities: brushing teeth, hitting, knocking, working at a PC, running, walking, and swinging. The system reports an average response time of less than 10 ms, which supports its feasibility. All the computations are done in an embedded system that should be carried by the user as an additional device. This has some disadvantages with respect to a mobile phone in terms of portability, comfort, and cost. Moreover, the size of the time window was chosen to be 160 ms. Given the nature of the recognized activities, this excessive granularity causes accidental movements when swinging or knocking to be confused with running, for instance. Such a small window length also (1) induces more overhead due to the classification algorithm being triggered very often and (2) is not beneficial for the feature extraction performance as time domain features require $O(n)$ computations.

2.3.1.7 Other Approaches

The system proposed by Brezmes et al. [28] features a mobile application for HAR under the Nokia platform. They used the $k$-nearest neighbors classifier, which is computationally expensive and not scalable for mobile phones as it needs the entire training set —that can be fairly large— to be stored in the device. Besides, their system requires each new user to collect additional training data in order to obtain accurate results. Ermes et al. [51] developed an online system that reaches 94% overall average accuracy but they only applied
2.3.1.8 Discussion

Each online HAR system has its own benefits and drawbacks. Thus, the selection of a particular approach for a real case study depends on the application requirements. If portability and obtrusiveness are the key issues, eWatch would be an appropriate option for ambulation activities. Nonetheless, if a broader set of activities needs to be recognized, COSAR should be considered, although it entails higher energy expenditures due to Bluetooth communication and the GPS sensor. The system proposed by Tapia et al. would be a better choice to monitor exercise habits yet it may be too obtrusive. Overall, most systems exhibit similar accuracy levels (more than 92%), but since each one works with a specific dataset and activity set, there is no significant evidence to argue that a system is more accurate than the others. Vigilante is the only approach that collects vital sign information, which opens a broader spectrum of applications for healthcare purposes. In addition, COSAR and Vigilante work under the Android platform —reported as best-selling smartphone platform in 2010 by Canalys [81]— facilitating the deployment in current cellular phones. The system cost is also an important aspect, especially when the application’s aim is being scaled to hundreds or thousands of users. Vigilante, COSAR, eWatch, and the work of Kao et al. require specialized hardware such as sensors and embedded computers whereas ActiServ and the system proposed by Brezmes et al. only need a conventional cellular phone.

2.3.2 Supervised Offline Systems

There are cases in which the user does not need to receive immediate feedback. For example, applications that analyze exercise and diet habits in patients with heart disease, diabetes, or obesity, as well as applications that estimate the number of calories burned after an exercise routine [82, 83] can work on an offline basis. Another example is the discovery of
user commercial patterns for advertisement. For instance, if an individual performs exercise activities very frequently, they could be advertised on sport wear items. In all these cases, gathered data can be analyzed on a daily —or even weekly— basis to draw conclusions on the person’s behavior. Table 2.8 summarizes state-of-the-art works in supervised offline human activity recognition based on wearable sensors. The most relevant approaches are described next.

2.3.2.1 Parkka

The work of Parkka et al. [16] considers seven activities: lying, rowing, riding a bike, standing still, running, walking, and Nordic walking. Twenty two signals were measured, including acceleration, vital signs, and environmental variables. This requires a number of sensors on the individual’s chest, wrist, finger, forehead, shoulder, upper back, and armpit. The integration device is a compact computer placed in a 5 kg rucksack. Therefore, this system was catalogued as highly obtrusive. Time- and frequency-domain features were extracted from most signals while a speech recognizer [56] was applied to the audio signal. This entails not only high processing demands but also privacy issues due to continuous recording of the user’s speech. Three classification methods were evaluated, namely an automatically generated decision tree, a custom decision tree which introduces domain knowledge and visual inspection of the signals, as well as an artificial neural network. The results indicate that the highest accuracy was 86%, given by the first method, though activities such as rowing, walking, and Nordic walking were not accurately discriminated. Parkka et al. mentioned that one of the causes of such misclassification is the lack of synchronization between the activity performances and annotations.

2.3.2.2 Bao

With more than 700 citations [84], the work of Bao and Intelle in 2004 [23] brought significant contributions to the field of activity recognition. The system recognizes 20 activities, including ambulation and daily activities such as scrubbing, vacuuming, watching
TV, and working at the PC. All the data were labeled by the user in a naturalistic environment. Five bi-axial accelerometers were initially placed on the user's knee, ankle, arm, and hip, yet they concluded that with only two accelerometers —on the hip and wrist— the recognition accuracy is not significantly diminished (in about a 5%). Using time- and frequency-domain features along with the C4.5 decision tree classifier, the overall accuracy was 84%. Ambulation activities were recognized very accurately (with up to 95% of accuracy) but activities such as stretching, scrubbing, riding escalator and riding elevator were often confused. The inclusion of location information is suggested to overcome this issue. This idea was later adopted and implemented by other systems [27].

2.3.2.3 Khan

The system proposed by Khan et al. [45] not only recognizes ambulation activities, but also transitions among them, e.g., sitting to walking, sitting to lying, and so forth. An accelerometer was placed on the individual’s chest, sampling at 20Hz, and sending data to a computer via Bluetooth for storage. Three groups of features were extracted from the acceleration signals: (1) autoregressive model coefficients, (2) the Tilt Angle (TA), defined as the angle between the positive Z-axis and the gravitational vector g, as well as the (3) Signal Magnitude Area (SMA), which is the summation of the absolute values of all three signals. Linear Discriminant Analysis was used to reduce the dimensionality of the feature vector and an Artificial Neural Network classified activities and transitions with a 97.76% subject independent accuracy. The results indicate that the TA plays a key role in the improvement of the recognition accuracy. This is expected because the sensor inclination values are clearly different for lying and standing, in view of the sensor being placed on the chest. One of the drawbacks of this system is its high computational complexity as it requires noise reduction, state recognition, time and frequency-domain features, LDA, and an a neural network to recognize activities.
Table 2.8: Summary of state-of-the-art in offline human activity recognition systems.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Activities</th>
<th>Sensors</th>
<th>Obtrusive</th>
<th>ID</th>
<th>Experiment</th>
<th>Flexibility</th>
<th>Features</th>
<th>Learning</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bao [23]</td>
<td>AMB, DA (20)</td>
<td>ACC (wrist, ankle, thigh, elbow, hip)</td>
<td>High</td>
<td>None</td>
<td>NAT</td>
<td>MNL</td>
<td>TD, FD</td>
<td>KNN, C4.5, NB</td>
<td>84%</td>
</tr>
<tr>
<td>Hanai [24]</td>
<td>AMB (5)</td>
<td>ACC (chest)</td>
<td>Low</td>
<td>Laptop</td>
<td>N/S</td>
<td>MNL</td>
<td>HAAR filters</td>
<td>C4.5</td>
<td>93.91%</td>
</tr>
<tr>
<td>Parkka [16]</td>
<td>AMB, DA (9)</td>
<td>ACC, ENV, VS (22 signals)</td>
<td>High</td>
<td>PC</td>
<td>NAT</td>
<td>MNL</td>
<td>TD, FD</td>
<td>DR, KNN</td>
<td>86%</td>
</tr>
<tr>
<td>He [25]</td>
<td>AMB (4)</td>
<td>ACC</td>
<td>Low</td>
<td>PC</td>
<td>N/S</td>
<td>MNL</td>
<td>AR</td>
<td>SVM</td>
<td>92.25%</td>
</tr>
<tr>
<td>He [41]</td>
<td>AMB (4)</td>
<td>ACC (trousers pocket)</td>
<td>Low</td>
<td>PC</td>
<td>N/S</td>
<td>MNL</td>
<td>DCT, PCA</td>
<td>SVM</td>
<td>97.51%</td>
</tr>
<tr>
<td>Zhu [67]</td>
<td>AMB, TR (12)</td>
<td>ACC (wrist, waist)</td>
<td>High</td>
<td>PC</td>
<td>N/S</td>
<td>SPC</td>
<td>AV, 3DD</td>
<td>HMM</td>
<td>90%</td>
</tr>
<tr>
<td>Altun [55]</td>
<td>AMB (19)</td>
<td>ACC, GYR (chest, arms, legs)</td>
<td>High</td>
<td>None</td>
<td>NAT</td>
<td>MNL</td>
<td>PCA, SFFS</td>
<td>BN, LS, KNN, DTW, ANN LDA</td>
<td>87% - 99%</td>
</tr>
<tr>
<td>Cheng [39]</td>
<td>UB (11)</td>
<td>Electrodes (neck, chest, leg, wrist)</td>
<td>High</td>
<td>PC</td>
<td>LAB</td>
<td>MNL</td>
<td>TD</td>
<td>SMCRF</td>
<td>88.38%</td>
</tr>
<tr>
<td>McGlynn [60]</td>
<td>DA (5)</td>
<td>ACC (thigh, hip, wrist)</td>
<td>Low</td>
<td>None</td>
<td>N/S</td>
<td>SPC</td>
<td>DTW</td>
<td>DTW ensemble</td>
<td>84.3%</td>
</tr>
<tr>
<td>Pham [61]</td>
<td>AMB, DA (4)</td>
<td>ACC (jacket)</td>
<td>Medium</td>
<td>N/S</td>
<td>N/S</td>
<td>Both</td>
<td>Relative Energy</td>
<td>NB, HMM</td>
<td>97% (SD), 95% (SI)</td>
</tr>
<tr>
<td>Vinh [68]</td>
<td>AMB, DA (21)</td>
<td>ACC (wrist, hip)</td>
<td>Medium</td>
<td>N/S</td>
<td>N/S</td>
<td>N/S</td>
<td>TD</td>
<td>SMCRF</td>
<td>88.38%</td>
</tr>
<tr>
<td>Centinela [33]</td>
<td>AMB (5)</td>
<td>ACC and VS (chest)</td>
<td>Medium</td>
<td>Cellphone</td>
<td>NAT</td>
<td>MNL</td>
<td>TD, FD, PR, TF</td>
<td>ALR, Bagging, C4.5, NB, BN</td>
<td>95.7%</td>
</tr>
<tr>
<td>Khan [45]</td>
<td>AMB, TR (15)</td>
<td>ACC (chest)</td>
<td>Medium</td>
<td>Computer</td>
<td>NAT</td>
<td>MNL</td>
<td>AR, SMA, TA, LDA</td>
<td>ANN</td>
<td>97.9%</td>
</tr>
<tr>
<td>Jatoba [17]</td>
<td>AMB (6)</td>
<td>ACC, SPI</td>
<td>High</td>
<td>Tablet</td>
<td>LAB</td>
<td>Both</td>
<td>TD / FD</td>
<td>CART, KNN</td>
<td>86% (SI), 95% (SD)</td>
</tr>
<tr>
<td>Chen [42]</td>
<td>AMB, DA, HW (8)</td>
<td>ACC (2 wrists)</td>
<td>Medium</td>
<td>N/S</td>
<td>LAB</td>
<td>MNL</td>
<td>TD, FD</td>
<td>FBF</td>
<td>93%</td>
</tr>
<tr>
<td>Minnen [59]</td>
<td>AMB, MIL (14)</td>
<td>ACC (6 places)</td>
<td>High</td>
<td>Laptop</td>
<td>Both</td>
<td>SPC</td>
<td>TD, FD</td>
<td>Boosting</td>
<td>90%</td>
</tr>
</tbody>
</table>
2.3.2.4 Zhu

The system proposed by Zhu and Sheng [67] uses *Hidden Markov Models* (HMM) to recognize ambulation activities. Two accelerometers, placed on the subject’s wrist and waist, are connected to a PDA via serial port. The PDA sends the raw data via Bluetooth to a computer which processes the data. This configuration is obtrusive and uncomfortable as the user has to wear wired links that may interfere with the normal course of activities. The extracted features are the angular velocity and the 3D deviation of the acceleration signals. The classification of activities operates in two stages. In the first place, an Artificial Neural Network discriminates among stationary (e.g., sitting and standing) and non-stationary activities (e.g., walking and running). Then, a HMM receives the ANN’s output and generates a specific activity prediction. An important issue related to this system is that all the data were collected from one single individual, which does not permit to draw strong conclusions on the system flexibility.

2.3.2.5 Centinela

This dissertation presents Centinela, a system that combines acceleration data with vital signs to achieve accurate activity recognition. Centinela recognizes five ambulation activities and includes a portable and unobtrusive real-time data collection platform, which only requires a single sensing device and a mobile phone. Time- and frequency-domain features are extracted from acceleration signals while polynomial regression and transient features [33] are applied to physiological signals. After evaluating eight different classifiers and three different time window sizes, and six feature subsets, Centinela achieves over to 95% overall accuracy. The results also indicate that incorporating physiological signals allows for a significant improvement of the classification accuracy. As a tradeoff, Centinela relies on classifier ensembles accounting higher computational cost, and it requires wireless communication with an external sensor, increasing energy expenditures.
2.3.2.6 Other Approaches

In 2002, Randel et al. [65] introduced a system to recognize ambulation activities which calculates the Root Mean Square (RMS) from acceleration signals and makes use of a Backpropagation Neural Network for classification. The overall accuracy was 95% using user-specific training but no details are provided regarding the characteristics of the subjects, the data collection protocol, and the confusion matrix. The system proposed in [24] uses HAAR filters to extract features and the C4.5 algorithm for classification purposes. HAAR filters are intended to reduce the feature extraction computations, compared to traditional TD and FD features. However, the study only collected data from four individuals with unknown physical characteristics, which might be insufficient to provide flexible recognition of activities on new users. He et al. [40, 25, 41] achieved up to 97% of accuracy but only considered four activities: running, being still, jumping, and walking. These activities are quite different in nature, which considerably reduces the level of uncertainty thereby enabling higher accuracy. Chen et al. [42] introduces an interesting Dynamic LDA approach to add or remove activity classes and training data online, i.e., the classifier does not have to be re-trained from scratch. With a Fuzzy Basis Function classifier, they reach 93% of accuracy for eight ambulation and daily activities. Nonetheless, all the data were collected inside the laboratory, under controlled conditions. Finally, Vinh et al. [68] use semi-Markovian conditional random fields to recognize not only activities but routines such as dinner, commuting, lunch, and office. These routines are composed by sequences of subsets of activities from a total set of 20 activities. Their results indicate 88.38% of accuracy (calculated by the authors from the reported recall tables).

2.3.2.7 Discussion

Unlike online systems, offline HAR are not dramatically affected by processing and storage issues because the required computations could be done in a server with huge computational and storage capabilities. Additionally, energy expenditures are not analyzed in
detail as a number of systems require neither integration devices nor wireless communication so the application lifetime would only depend on the sensor specifications.

Ambulation activities are recognized very accurately by [85, 45, 33]. These systems place an accelerometer on the subject’s chest, which is helpful to avoid ambiguities due to abrupt corporal movements that arise when the sensor is on the wrist or hip [58]. Other daily activities such as dressing, preparing food, using the bathroom, using the PC, and using a phone are considered in [60]. This introduces additional challenges given that, in reality, an individual could use the phone while walking, sitting, or lying, thereby exhibiting different acceleration patterns. Similarly, in [23], activities such as eating, reading, walking, and climbing stairs could happen concurrently yet no analysis is presented to address that matter. Chapter 6 provides insights to the problem of recognizing concurrent activities.

Unobtrusiveness is a desirable feature of any HAR system but having more sensors enables the recognition of a broader set of activities. The scheme presented by Cheng et al. [39] recognizes head movements and activities such as swallowing, chewing, and speaking but requires obtrusive sensors on the throat, chest, and wrist, connected via wired links. In tactical scenarios, this should not be a problem considering that a soldier is accustomed to carry all sort of equipment (e.g., sensors, cameras, weapons, and so forth). Yet, in healthcare applications involving elderly people or patients with heart disease, obtrusive sensors are not convenient.

The studies presented in [42, 39, 17] are based on data collected under controlled conditions while the works in [68, 61, 60, 40, 25, 41] do not specify the data collection procedure. This is a critical issue since a laboratory environment affects the normal development of human activities [23, 38]. The number of subjects also plays a significant role in the validity of any HAR study. In [67, 59] only one individual collected data while in [24], data were collected from four individuals. Collecting data from a small number of people might be insufficient to provide flexible recognition of activities on new users.
2.3.3 Semi-supervised Approaches

The systems studied so far rely on large amounts of labeled training data. Nonetheless, in some cases, labeling all instances may not be feasible. For instance, to ensure a naturalistic data collection procedure, it is recommended for users to perform activities without the participation of researchers. If the user cannot be trusted or the activities change very often, some labels could be missed. These unlabeled data can still be useful to train a recognition model by means of semi-supervised learning. Some of the most important works in this field are described next.

2.3.3.1 Multi-graphs

Stikic et al. [74, 75] developed a multi-graph-based semi-supervised learning technique which propagates labels through a graph that contains both labeled and unlabeled data. Each node of the graph corresponds to an instance while every edge encodes the similarities between a pair of nodes as a probability value. The topology of the graph is given by the $k$-nearest neighbors in the feature space. A probability matrix $Z$ is estimated using both Euclidean distance in the feature space and temporal similarity [74]. Once the labels have been propagated throughout the graph (i.e., all instances are labeled), classification is carried out with a Support Vector Machine classifier that relies on a Gaussian radial basis function kernel. The classifier also used the probability matrix $Z$ to introduce knowledge on the level of confidence of each label. The overall accuracy was up to 89.1% and 96.5% after evaluating two public datasets and having labels for only 2.5% of the training data.

2.3.3.2 En-co-training

A well-known method in semi-supervised learning is co-training, proposed by Blum and Mitchel in 1998 [86]. This approach requires the training set to have two sufficient and redundant attribute subsets, condition that does not always hold in a HAR context. Guan et al. [77] proposed en-co-training, an extension of co-training which does not have the limitations of its predecessor. The system was tested with ten ambulation activities and
compared to three other fully supervised classifiers (the \( k \)-nearest neighbors, naïve Bayes and a decision tree). The maximum error rate improvement reached by en-co-training was from 17\% to 14\% —when 90\% of the training data were not labeled. If 20\% or more of the training data are labeled, the error rate difference between en-co-training and the best fully supervised classifier does not exceed 1.3\%.

### 2.3.3.3 Ali

Ali et al. [76] implemented a Multiple Eigenspaces (MES) technique based on the Principal Component Analysis combined with Hidden Markov Models. The system is designed to recognize finger gestures with a laparoscopic gripper tool. The individuals wore a sensor glove with two bi-axial accelerometers sampling at 50Hz. Five different rotation and translation movements from the individual’s hand were recognized with up to 80\% of accuracy. This system becomes hard to analyze since no details are provided on the amount of labeled data nor the evaluation procedure.

### 2.3.3.4 Huynh

Huynh et al. [78] combined Multiple Eigenspaces with Support Vector Machines to recognize eight ambulation and daily activities. Eleven accelerometers were placed on individuals' ankles, knees, elbows, shoulders, wrists, and hip. The amount of labeled training data varied from 5\% to 80\% and the overall accuracy was between 88\% to 64\%, respectively. Their approach also outperformed the fully supervised naïve Bayes algorithm, which was used as a baseline. Still, activities such as shaking hands, ascending stairs and descending stairs were often confused.

### 2.3.3.5 Discussion

The next step in semi-supervised learning HAR would be their implementation online, opening the possibility to use the data collected in production stage —which are unlabeled—to improve the recognition performance. Nevertheless, implementing this approach becomes
challenging in terms of computational complexity. This is because most semi-supervised HAR approaches first estimate the labels of all instances in the training set and then apply a conventional supervised learning algorithm. Additionally, the label estimation process is often computationally expensive; for instance, in [74, 75], a graph with one node per instance has to be built. In their experiments, the resulting graphs consisted of up to 16875 nodes, causing the computation of the probability matrix to be highly demanding in regards to processing and storage. Other approaches do not seem to be ready for real scenarios: En-co-training [77] did not report substantial improvement in the classification accuracy. The system proposed by Ali et al. [76] was intended for a very specific purpose but not suitable for recognizing daily activities thereby limiting its applicability to context-aware applications. Finally, the system proposed in [78] required eleven sensors, which introduces high obtrusiveness. Overall, the field of semi-supervised activity recognition has not reached maturity and needs additional contributions to overcome the aforementioned issues.

2.4 Multiple Classifier Systems

Pattern classification has been a very productive research area in past years. Innumerable applications can be visualized in forecasting, speech recognition, image processing, and bioinformatics, among others. In most cases, a single classification model is trained and evaluated fixing its parameters to maximize the classification accuracy. Still, selecting the best classification algorithm for a given dataset is not always an easy task. Even though cross validation and statistical hypothesis testing (e.g., 5 × 2-fold cross validation with a paired t-test) are often used to perform such selection, there are cases where no significant evidence can be found to assure that a classifier is better than another. Then, considering predictions not from one, but from a set of classifiers, turns out to be a good alternative to enrich a pattern recognition system. This is the main idea behind multiple classifier systems (MCS’s), which rely on the hypothesis that a set of base classifiers (i.e., the individual experts part of the MCS) may provide more accurate and diverse predictions [87]. Of course, MCS’s entail additional complexity, not only because a number of classifiers
should be trained and evaluated, but also because they require defining criteria to generate a prediction from their outputs.

2.4.1 Types of Multiclassifier Systems

Although there is no consensus on the taxonomy of multiclassifier systems, most authors agree with four levels for building classifier ensembles:

- In the data level, the most noticeable methods are bagging and boosting. Bagging uses k identical classifiers, each with a different bag—a subsample of the training dataset. The bags are generally overlapping and the final decision is generated by voting. Instead, boosting iteratively builds a new classifier, assigning more weight to the instances that were incorrectly classified. In that way, new models become experts in feature subspaces where earlier models were not successful. As does bagging, boosting also uses a voting principle to output the final prediction.

- In the feature level, feature selection algorithms discussed in Section 2.2.1.5 yield different feature subsets which could be the input of each of the base classifiers in the ensemble.

- In the classifier level, different classification algorithms could be applied. These should be carefully selected since redundant or inaccurate classifiers might degrade the ensemble performance. Thus, quantitative criteria such as the correlation [88] and disagreement [89] are suggested practices to assure a successful MCS. In this dissertation, we also propose an algorithm to select base classifiers based on the concept of level of collaboration.

- Finally, in the combiner level, the main goal is to define a mechanism to estimate the correct label given a set of predictions from the base classifiers. This problem, also addressed in Chapter 5, is more rigorously formulated in Definition 3.
This dissertation focuses on the design of new combination and classifier level strategies in multiple classifier systems. Hence, the rest of this section is intended to cover the most relevant approaches in these categories.

### 2.4.2 Classifier-level Approaches

A successful multiclassifier system should maintain base classifiers with high diversity. If the base classifiers are rather redundant, the ensemble would not deliver significant improvement. This is a hard problem to be solved deterministically as for $k$ possible base classifiers, there are $O(2^k)$ possible subsets. Therefore, a variety of heuristic methods based on diversity metrics have been proposed. Two of the most common metrics are the correlation [88] and the disagreement [89].

**Definition 4 (Classifier correlation)** The pairwise correlation $\rho_{ij}$ between classifiers $D_i$ and $D_j$ is defined as follows:

$$\rho_{ij} = \frac{ad - bc}{\sqrt{(a + b)(c + d)(a + c)(b + d)}}$$  \hspace{1cm} (2.14)

where:

- $a$ is the number of instances correctly classified by both classifiers.
- $b$ is the number of instances correctly classified by classifier $D_i$ but incorrectly classified by classifier $D_j$.
- $c$ is the number of instances correctly classified by classifier $D_j$ but incorrectly classified by classifier $D_i$.
- $d$ is the number of instances incorrectly classified by both classifiers.

**Definition 5 (Disagreement level)** The disagreement level between classifiers $D_i$ and $D_j$ is defined as follows:

$$R(D_i, D_j) = \frac{b + c}{a + b + c + d}$$  \hspace{1cm} (2.15)

47
Given these definitions, the idea is to choose a set of classifiers with low correlation values and high disagreement levels, thereby inducing more diversity in the ensemble. However, the combiner could be affected if diverse yet inaccurate classifiers are chosen. Section 5.2.4 provides a deeper understanding of this matter.

2.4.3 Combination-level Approaches

There exist two main approaches to combine multiple base classifiers, namely fusion and selection [90]. The former works under the assumption that all base classifiers are competent in the entire feature space. Thus, voting and averaging are common methods in classifier fusion. In the latter, each classifier is assumed to be an expert in a certain feature subspace so the selection of a classifier usually requires to explore the neighborhood of the instance to be classified. In general, classifier fusion has been the focus of more profound research. On the other hand, according to Zhu et al. [91], classifier selection might be either static (i.e., the classifier is selected at training time) or dynamic (i.e., the classifier is selected at evaluation time). Dynamic selection usually yields more accurate predictions, yet it also entails higher computational complexity than static selection.

2.4.3.1 Classifier Fusion

The very first and most intuitive approaches in decision fusion apply voting. Some examples are the simple majority (i.e., returning the prediction with more than a half of the votes) and the plurality (i.e., returning the prediction with the highest number of votes) [92]. Despite their simplicity, these have been shown to be effective in many learning problems [91]. However, if the accuracies of the base classifiers are very dissimilar, these two approaches may not be beneficial as each base classifier has the same impact on the final decision. Weighted majority vote [87] was then introduced to overcome such issue by assigning a weight to each classifier according to estimations of its prediction probabilities.

A different approach consists of estimating probabilities for each class using the confusion matrix. This is the main philosophy behind the naïve Bayes combination [87], which
assumes conditional independence among the classifier predictions \( S = \{ s_0, ..., s_{k-1} \} \) given a class \( \omega_i \). The probability of each class is calculated as follows:

\[
P(\omega_j | S) \propto P(\omega_j) \prod_{i=0}^{k-1} P(\omega_j | s_i)
\] (2.16)

Then, the class with the highest probability is selected as the ensemble’s prediction. One of the drawbacks of this method is that, if a classifier performs poorly on certain prediction, it would highly affect the output of the ensemble.

### 2.4.3.2 Classifier Selection

In classifier selection, the goal is to design a mechanism to always (or at least in most of the cases) choose the best classifier for a given input. Generally, this is achieved by estimating the posterior probability (i.e., the probability that a classifier’s output is correct given a class) for each prediction in local regions of the feature space. Such regions can be determined by the classes themselves or could be defined by a vicinity criterion.

In the first case, the aim is to find the probability \( P(\omega^* = \omega_j | s_i = \omega_j) \) that a prediction \( s_i = D_i(x) \), corresponding to class \( \omega_j \), matches the correct class \( \omega^* \). In virtue of the Bayes’ theorem, Giacinto et al. [93] have expressed this probability as follows:

\[
\hat{P}(\omega^* = \omega_j | s_i = \omega_j) = \frac{\hat{P}(s_i = \omega_j | \omega^* = \omega_j) \hat{P}(\omega^* = \omega_j)}{\hat{P}(s_i = \omega_j)}
\] (2.17)

Now, such probabilities could be estimated from the confusion matrix:

\[
\hat{P}(\omega^* = \omega_j | s_i = \omega_j) = \frac{TP_{ij} \cdot TP_{ij} + FN_{ij}}{TP_{ij} + FP_{ij}} = \frac{TP_{ij}}{TP_{ij} + FP_{ij}}
\] (2.18)

where True Positives \( (TP_{ij}) \), False Positives \( (FP_{ij}) \) and False Negatives \( (FN_{ij}) \), are from the evaluation of classifier \( D_i \) on class \( \omega_j \). Observe this result is nothing but the precision or positive predictive value of classifier \( y_j \) for class \( i \). Using the values in the confusion matrix, Equation 2.18 can be rewritten as follows:
\[
\hat{P}(\omega^* = \omega_j \mid s_i = \omega_j) = \frac{M^i_{\omega_j \omega_j}}{\sum_{r=0}^{n-1} M^i_{r \omega_j}}
\]  

(2.19)

where \(M^i_{n \times n}\) is the confusion matrix for classifier \(D_i\).

Now, in the second case —vicinities as local regions— Woods et al. [94] proposed a dynamic classifier selection approach via local accuracy estimates (DCS-LA). Given an unknown instance \(x\) to be classified, the ensemble’s output is given by the most accurate classifier in a local region defined by the \(k\) nearest neighbors of \(x\) in the training set. Later, other approaches adopted this methodology, proposing several improvements and extensions [95, 87, 96]. Nonetheless, this family of methods exhibit a high computational cost as each new instance to be classified should be compared to the entire training dataset —which could be fairly large. Moreover, this approach is very sensitive to noise and is highly biased by the size of the neighborhood, which is usually problem-dependent.

Hybrid methodologies have also been subject of study. As an example, the approach proposed by Cavalin et al. [97] applies the concept of multistage organization for dynamic classifier selection yet it is computationally expensive, requiring a hundred classifiers with different samples of the dataset and a genetic algorithm to select a prediction.
Chapter 3: Physiological Signals in Activity Recognition

3.1 Note to the Reader

Part of this chapter was published in the Elsevier Pervasive and Mobile Computing [33]. Appendix A includes the permissions to reuse such work in this dissertation. The corresponding article “Centinela: A Human Activity Recognition System based on Acceleration and Vital Sign Data” may be found in Appendix B.

3.2 Introduction

As it was mentioned in Chapter 2, most of the previously proposed activity recognition schemes collect data from either triaxial accelerometers, video sequences [9], or environmental variables. However, little work has been reported considering vital sign data. It is not difficult to show that there is a noticeable relationship between the behavior of physiological signals and the user’s activity. When an individual begins running, for instance, it is expected that their heart rate and breath amplitude increase. Consequently, the hypothesis of this chapter is that higher human activity recognition accuracy can be achieved using both acceleration and vital sign data. To illustrate this, consider the situation in Figure 3.1. Data from triaxial acceleration and vital signs were recorded while a subject was ascending (i.e., walking upstairs) after walking. Note that the acceleration signals within most time intervals are very similar for both activities. Instead, the heart rate time series exhibits a very clear pattern, as a person requires more physical effort to climb stairs than to walk. This might allow to classify said activities more accurately.

This chapter presents Centinela, a human activity recognition system which considers acceleration and physiological signals. The proposed methodology encompasses (1) collect-
Figure 3.1: Acceleration signals and heart rate for the activities walking and ascending.

The main features of Centinela are listed below:

- Centinela combines acceleration data with vital signs to achieve highly accurate activity recognition. In fact, it provides higher accuracy than using acceleration signals solely.

- Five activities are recognized as a proof of concept: walking, running, being still (sitting or standing), ascending (i.e., walking upstairs), and descending (i.e., walking downstairs).

- Since vital signs are not expected to change abruptly, Centinela applies structure detectors [54], i.e., linear and non-linear functions, to extract features.

- Three new features were proposed for physiological signals: trend, magnitude of change, and signed magnitude of change, intended to discriminate among activities during periods of vital sign stabilization.
• Centinela relies on a portable and unobtrusive real-time data collection platform, which allows not only for activity recognition but also for monitoring health conditions of target individuals.

• Several classifiers are analyzed in this study, allowing other researchers and application developers to use the most appropriate classifiers for specific activities.

The rest of the chapter is organized as follows: Section 3.3 introduces the global structure of Centinela. Section 3.3.1 describes the data acquisition architecture, as well as the data collection protocol. Section 3.3.2 covers the methods applied for feature extraction, i.e., statistical, structural, and transient features. Then, Section 3.4 presents the methodology of the experiments and main results. Finally, Section 3.5 summarizes the most important conclusions and findings.

3.3 Description of the System

Figure 3.2 illustrates the process fulfilled for activity recognition. First, labeled data are collected from accelerometer and vital sign sensors, as described in Section 3.3.1. Then, time- and frequency-domain statistical feature extraction are applied to the acceleration signals (Section 3.3.2.1), as well as structural and transient features are extracted from vital signs (Sections 3.3.2.2 and 3.3.2.3). Next, the dataset with the extracted features is passed as input to various classification algorithms in order to select the most appropriate model (Section 3.4).

3.3.1 Data Collection

Figure 3.3 shows the system architecture for the data collection phase. The sensing device (see Section 3.3.1.1 for more details) communicates via Bluetooth with an Internet-enabled cellphone. There is a mobile application which decodes the packets and sends labeled data to the application server via Internet. The server then receives these data and stores them into a relational database.
3.3.1.1 Sensing Device

Centinela uses the BioHarness BT™ chest sensor strap [80] manufactured by Zephyr Technology (see Figure 4.2). This device features a triaxial accelerometer and allows for measuring vital signs as well. The strap is unobtrusive, lightweight, and can be easily worn by any person. The measured attributes are: heart rate (i.e., pulse), respiration rate, breath waveform amplitude, skin temperature, posture (i.e., inclination of the sensor), ECG amplitude, and 3D acceleration, among others. The accelerometer records measurements at 50 Hz, each one between -3g to 3g, where g stands for the acceleration of gravity. Acceleration samples are aggregated in packets sent every 400 ms, so every packet contains twenty acceleration measurements in all three dimensions. On the other hand, the vital signs are sampled at 1 Hz., since they are not expected to change considerably in short periods of time.

In the literature, accelerometers are commonly placed on the wrist [23, 51, 16], ankle [23, 51], or in the trouser’s pocket [25, 41, 40], yet a person might be, for instance, moving his/her arms or legs while been seated. This fact may introduce noise to the data, thereby causing misclassification. Therefore, placing the accelerometer on the chest makes the system more noise tolerant, and the results presented in Section 3.4.2 support this hypothesis.
Figure 3.3: Data collection architecture.

Figure 3.4: Mobile application user interface [1].
Table 3.1: Physical characteristics of the participants.

<table>
<thead>
<tr>
<th></th>
<th>Avg</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (years)</td>
<td>24</td>
<td>9</td>
<td>34</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>76.5</td>
<td>27</td>
<td>95</td>
</tr>
<tr>
<td>Height (m)</td>
<td>1.74</td>
<td>1.35</td>
<td>1.88</td>
</tr>
<tr>
<td>BMI (kg/m²)</td>
<td>24.23</td>
<td>20.96</td>
<td>29</td>
</tr>
</tbody>
</table>

3.3.1.2 Mobile Application

A mobile software application was built to collect training data under the Java ME platform. This allows Centinela to run in any mobile phone that supports Java, thereby avoiding the inconvenience of requiring the user to carry additional recording devices. The mobile application receives and decodes the raw data sent from the sensor via Bluetooth, visualizes the measurements (see Figure 3.4.A), and labels each measurement according to the option selected by the user, either: running, walking, sitting, ascending, or descending (see Figure 3.4.B). The samples are sent in real time, via UDP, to the application server, which stores the labeled data into a relational PostgreSQL database. As Java ME is falling into disuse, the mobile component of Centinela was migrated to the Android platform, including additional features and functionalities. In Chapter 4, the reader may find more details about this matter.

3.3.1.3 Data Collection Protocol

The data were collected in a naturalistic fashion, thus, no specific instructions were given to the participants on how to perform the activities. The speed, intensity, gait, and other environmental conditions were arbitrarily chosen by the subjects. Eight individuals, 7 males and 1 female, participated in this study. Their physical characteristics, namely age, weight, height, and body mass index are summarized in Table 3.1.

Unlike accelerometer signals, vital signs do not abruptly vary after the person changes activities. On the contrary, the values of vital signs during time interval $I_j$ depend of the activity during $I_{j-1}$. If the individuals were at rest before recording each session, the
system would not be trained to recognize interleaving activities! Consequently, the data were collected from subjects while performing successive pairs of activities, e.g., running before sitting, walking before descending, and so on. This was carried out for all twenty possible combinations of pairs of consecutive activities.

3.3.2 Feature Extraction

In general, two approaches have been proposed to extract features in time series data: statistical and structural [54]. The former, such as the Fourier transform and the Wavelet transform, use quantitative characteristics of the data to extract features. The latter take into account the morphological interrelationship among data. Hence, they have been widely used for image processing and time series analysis. Due to both acceleration and physiological signals being distinct in nature, Centinela applies methods from both statistical and structural feature extraction.

Now, to overcome the problem of detecting transitions between activities, all measured signals were divided into fixed size 50% overlap time windows, as suggested by [42, 23]. For every time window, 84 features were extracted as follows: eight statistical features for each of the acceleration signals (i.e., 24 features), nine structural features for each of the physiological signals (i.e., 54 features), and one transient feature for each of the physiological signals (i.e., 6 features). Transient features are proposed in this dissertation (see Section 3.3.2.3) to address activity recognition during periods of vital sign stabilization —they could be considered structural as they also provide information on the signal shape. Table 3.2 summarizes the feature set computed from raw signals. The definitions of these features are presented in the following subsections.

3.3.2.1 Statistical Features from Acceleration Signals

Time-domain and frequency-domain features have been extensively used to filter relevant information within acceleration signals [42, 23, 22, 16, 20, 51, 21]. In this work, eight features were calculated for all three acceleration signals (a total of 24 features). These
Table 3.2: List of features extracted in this work.

<table>
<thead>
<tr>
<th>Measured Signals</th>
<th>Statistical</th>
<th>Structural</th>
<th>Transient</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccX (g)</td>
<td>×</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>AccY (g)</td>
<td>×</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>AccZ (g)</td>
<td>×</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Heart rate</td>
<td>×</td>
<td>×</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Respiration rate</td>
<td>×</td>
<td>×</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Breath amplitude</td>
<td>×</td>
<td>×</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Skin temperature</td>
<td>×</td>
<td>×</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Posture</td>
<td>×</td>
<td>×</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>ECG amplitude</td>
<td>×</td>
<td>×</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
<td>54</td>
<td>6</td>
<td>84</td>
</tr>
</tbody>
</table>

are: mean, variance, standard deviation, correlation between axes, interquartile range, mean absolute deviation, and root mean square, from the time domain; and, energy from the frequency domain. These features were defined in Section 2.2.1.1.

3.3.2.2 Structural Features from Physiological Signals

Previous works that explored vital sign data with the aim of recognizing human activities have applied statistical feature extraction. In [20], the authors computed the number of heart beats above the resting heart rate value as the only feature. Instead, Parkka et al. [16] calculated time domain features for heart rate, respiration effort, SaO$_2$, ECG, and skin temperature. Nevertheless, the signal’s shape is not described by these features. Consider the situation shown in Figure 3.5. A heart rate signal $S(t)$ for an individual that was walking is shown with a bold line and the same signal in reverse temporal order, $S'(t)$, is displayed with a thin line. Notice that most time domain and frequency domain features (e.g., mean, variance, and energy) are identical for both signals while they may represent different activities. This is the main motivation for rather applying structural feature extraction: describing the morphological interrelationship among data.

Given a time series $Y(t)$, a structure detector implements a function $f(Y(t)) = \hat{Y}(t)$ such that $\hat{Y}(t)$ represents the structure of $Y(t)$ as an approximation [54]. The extracted features are the parameters of $\hat{Y}(t)$, which depend on the nature of the function. In order
Figure 3.5: Heart rate signal for walking (bold) and flipped signal (thin).

Table 3.3: Common functions implemented by structure detectors.

<table>
<thead>
<tr>
<th>Function</th>
<th>Equation</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$\hat{Y}(t) = mt + b$</td>
<td>${m, b}$</td>
</tr>
<tr>
<td>Polynomial</td>
<td>$\hat{Y}(t) = a_0 + a_1t + \ldots + a_{n-1}t^{n-1}$</td>
<td>${a_0, \ldots, a_{n-1}}$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$\hat{Y}(t) = a</td>
<td>b</td>
</tr>
<tr>
<td>Sinusoidal</td>
<td>$\hat{Y}(t) = a \cdot \sin(t + b) + c$</td>
<td>${a, b, c}$</td>
</tr>
</tbody>
</table>

To measure the goodness of fit of $\hat{Y}(t)$ to $Y(t)$, the sum of squared errors (SSE) is calculated as follows:

$$SSE = \sum_i \left( Y(t) - \hat{Y}(t) \right)^2$$  \hspace{1cm} (3.1)

Then, for each measured attribute, the goal was to find the function $\hat{Y}^*(t)$ with the smallest SSE. Table 3.3 summarizes the different types of functions that have been evaluated in this work. The median of the SSE was calculated for all time windows from all six physiological signals and all four structure detectors. The median was preferred over the mean to prevent noisy samples to bias the goodness of fit of the feature detectors. From the evaluation, polynomial functions of third degree had the lowest SSE for all six vital signs. Polynomials of degree higher than three were not considered to avoid overfitting due to Runge’s phenomenon [98].
A total of nine structural features were extracted from each vital sign time window, i.e., the coefficients of the polynomials of degree one, two, and three that best fit the points in the time window.

3.3.2.3 Transient Features from Physiological Signals

Consider, for instance, that someone is running for one minute and then sits down for two minutes. Even though the individual is seated, their vital signs (e.g., heart rate, respiration rate, etc.) remain as if they was running for an interval of time called the transient period. To overcome this issue, three new features are proposed in this work —besides structural features—, namely the trend $\tau$, the magnitude of change $\kappa$, and the signed magnitude of change $\eta$, intended to describe the behavior of the vital signs during transient periods. The trend indicates whether the signal is increasing, decreasing, or constant. Notice that, due to the nature of the human activities considered in this work, it is expected that vital signs are either strictly increasing, strictly decreasing, or remain constant while an individual is performing one single activity.

**Definition 6 (Trend)** Let $m$ be the slope of the line that best fits the series $S$. Then, the trend $\tau(S, r)$ of $S$ is defined as follows:

$$
\tau(S, r) = \begin{cases} 
1 \text{ (increasing)} & \text{if } (m \geq r) \\
-1 \text{ (decreasing)} & \text{if } (m \leq -r) \\
0 \text{ (constant)} & \text{if } (|m| < |r|)
\end{cases} \tag{3.2}
$$

where $r$ is a positive real number that stands for the slope threshold.

This value was set to $\tan(15^\circ)$ after doing an experimental analysis over the entire the dataset. The trend can be computed in $O(1)$, given that the slope of the line that best fits the data points was calculated beforehand as one of the structural features.
Now, it is important not only to detect whether the vital signs increased or decreased in a time window, but also to measure how much they varied. For this purpose the magnitude of change feature is presented as follows:

**Definition 7 (Magnitude of change)** Let $S$ be a given time series defined from $t_{\text{min}}$ to $t_{\text{max}}$. Let $S_p^-$ be a subset of $S$ which contains all measurements between $t_{\text{min}}$ and $t_{\text{min}} + (t_{\text{max}} - t_{\text{min}})p$, where $0 < p < 1$ is a percentage of the series. Let $S_p^+$ be a subset of $S$ which contains all samples between $t_{\text{min}} + (t_{\text{max}} - t_{\text{min}})(1 - p)$ and $t_{\text{max}}$. Then, the magnitude of change $\kappa(S, p)$ is defined as

$$
\kappa(S, p) = \max \{ \max (S_p^+), \min (S_p^-) \},
\quad \kappa(S, p) = \max \{ \max (S_p^-) - \min (S_p^+), \min (S_p^-) - \min (S_p^+) \} \quad (3.3)
$$

![Figure 3.6: Calculation of the magnitude of change feature.](image)

The value of $p$ was set to 0.2 after doing an experimental analysis over the entire the dataset. This implies that $S^-$ is the first 20% of the series and $S^+$ would be the last 20% of the series. The purpose of the magnitude of change is to estimate the maximum deviation between the beginning and the end of the series, and it can be calculated in linear time. Figure 3.6 illustrates the process of calculating this feature.
Both magnitude of change and trend are combined in a single feature, the *signed magnitude of change* $\eta$ as follows:

**Definition 8 (Signed magnitude of change)**  *Given a time series $S$, the signed magnitude of change $\eta$ is defined as follows:*

\[
\eta(S,p,r) = \kappa(S,p)\tau(S,r)
\]  

(3.4)

where $p$ and $r$ are the parameters of $\kappa$ and $\tau$, respectively. Even though the transient features are strongly related to the parameters of a linear regression, they are different measures of the data shape. Section 3.4.2.4 analyzes the effectiveness of these proposed features.

### 3.4 Evaluation

This section describes the methodology to evaluate the system and provides further analysis and discussion of the main results and findings.

#### 3.4.1 Design of the Experiments

The recognition of activities was fulfilled by assessing two different datasets: the first one, $D_{acc}$, solely contains the features extracted from acceleration data; the second, $D_{vs}$, includes all features (i.e., statistical, structural, and transient). The comparison of these two datasets is with the purpose of measuring the impact of vital signs features in the classification accuracy. Seven classification algorithms were evaluated:

1. Naïve Bayes (NB) [71].

2. Bayesian Network (BN) using the K2 search algorithm [70].

3. J48 decision tree, which is an implementation of the C4.5 algorithm [53].
4. Multilayer Perceptron (MLP), which relies on a Backpropagation Neural Network [53].

5. Additive Logistic Regression (ALR) [99], performing Boosting with an ensemble of ten Decision Stump classifiers.

6. Bagging using an ensemble of ten Naïve Bayes classifiers (BNB) and each bag having the same size than the training set.

7. Bagging using an ensemble of ten J48 classifiers (BJ48) and each bag having the same size than the training set.

The interested reader may refer to [53, 70, 71, 99] for a complete description of these classification methods.

The evaluation encompasses two parts: the selection of the best classifier(s), and the calculation of their accuracy. In order to determine whether a classifier is better than another, a 5 \times 2-fold cross validation along with a paired \( t \)-test were performed, as suggested in [79]. In general, a two-fold cross validation is preferred to reduce the probability of concluding that classifier is better than another when it is not the case. For all the statistical tests, the significance level was fixed to \( \alpha = 0.05 \).

All the classification algorithms were tested in the Waikato Environment for Knowledge Analysis (WEKA) [30]. This is a well known software tool developed by the University of Waikato, New Zealand, which facilitates the evaluation and analysis of machine learning algorithms.

3.4.2 Results

An interesting fact in machine learning is that the performance of a classification algorithm depends on the dataset it is applied to. As the goal of this study is to proof that vital signs account for more accurate recognition, each classification algorithms was evaluated in both datasets.

The data repository used for this study is more complete than the one presented in [33]. In fact, the former has over 35 minutes of labeled data (i.e., up to 630 instances) while the
present one has only 19 minutes (i.e., up to 342 instances). Evidently, a larger dataset allows to draw more robust conclusions on the system performance. Two time window sizes were evaluated, namely 5s and 12s. In the previous study [33], longer time windows were shown to diminish the overall classification accuracy and could contain more than one activity.

In total, fourteen classifiers were evaluated—seven for each window size—with five different random seeds $s_i \in \{1, 128, 255, 1023, 4095\}$. As a notation, the name of the classifiers is accompanied by the window size written as a superscript and the dataset as a subscript. For example, $ALR_{vs}^{12s}$ stands for the additive logistic regression algorithm over the $D_{vs}$ dataset using 12s time windows.

### 3.4.2.1 Dataset with Features from Vital Signs and Acceleration

Table 3.4 shows the results of the 5 × 2-fold cross validation for the $D_{vs}$ dataset. Note that the highest overall accuracy was achieved by the $ALR_{vs}^{5s}$ classifier. In order to find the best classifier for this dataset, a paired two-tailed $t$-test was performed between the $ALR_{vs}^{5s}$ and all other classifiers with a significance level $\alpha = 0.05$. The null hypothesis is that each pair of classifiers achieved the same mean accuracy. As a result of the tests, the $p$-values are included in the last column of Table 3.4. Since all the $p$-values are below the significance level, there is strong statistical evidence that $ALR_{vs}^{5s}$ is more accurate than all other classifiers in the $D_{vs}$ dataset.

### 3.4.2.2 Dataset with Features from Acceleration Only

The same procedure was carried out over $D_{acc}$ (i.e., the dataset that only contains features from accelerometer data). Table 3.5 summarizes the results of the 5 × 2-fold cross validation for this dataset; here, the $BJ48_{acc}^{5s}$ classifier achieved the highest average accuracy (i.e., 88.7%). However, $ALR_{vs}^{5s}$ still outperforms all classifiers in $D_{acc}$ since the $p$-values are below $\alpha$. This is a very remarkable result as it provides strong statistical support for one of the hypothesis of this dissertation: physiological signals being beneficial to improve the recognition accuracy of human activities.
Table 3.4: Percentage classification accuracy given by the 5 × 2-fold cross validation on $D_{vs}$.

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>Avg.</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>88.87</td>
<td>90.78</td>
<td>89.35</td>
<td>88.71</td>
<td>88.08</td>
<td>89.16</td>
<td>0.001</td>
</tr>
<tr>
<td>NB</td>
<td>74.09</td>
<td>76.95</td>
<td>80.76</td>
<td>79.33</td>
<td>73.77</td>
<td>76.98</td>
<td>0.000</td>
</tr>
<tr>
<td>BN</td>
<td>90.62</td>
<td>90.14</td>
<td>89.67</td>
<td>89.83</td>
<td>88.08</td>
<td>89.67</td>
<td>0.008</td>
</tr>
<tr>
<td>ALR</td>
<td>91.57</td>
<td>93.64</td>
<td>93.64</td>
<td>93.48</td>
<td>93.32</td>
<td>93.13</td>
<td>-</td>
</tr>
<tr>
<td>J4</td>
<td>85.85</td>
<td>87.28</td>
<td>87.44</td>
<td>90.78</td>
<td>86.96</td>
<td>87.66</td>
<td>0.001</td>
</tr>
<tr>
<td>BNB</td>
<td>74.09</td>
<td>75.99</td>
<td>78.54</td>
<td>79.81</td>
<td>73.77</td>
<td>76.44</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>BJ4</td>
<td>87.28</td>
<td>92.53</td>
<td>88.39</td>
<td>92.21</td>
<td>90.14</td>
<td>90.11</td>
<td>0.021</td>
</tr>
<tr>
<td>MLP</td>
<td>80.14</td>
<td>83.39</td>
<td>82.67</td>
<td>85.56</td>
<td>80.51</td>
<td>82.45</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>NB</td>
<td>74.01</td>
<td>80.87</td>
<td>71.12</td>
<td>74.73</td>
<td>73.29</td>
<td>74.80</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>BN</td>
<td>85.20</td>
<td>89.53</td>
<td>85.56</td>
<td>88.09</td>
<td>88.81</td>
<td>87.44</td>
<td>0.009</td>
</tr>
<tr>
<td>ALR</td>
<td>90.25</td>
<td>92.06</td>
<td>91.70</td>
<td>90.97</td>
<td>91.34</td>
<td>91.26</td>
<td>0.001</td>
</tr>
<tr>
<td>J4</td>
<td>85.92</td>
<td>85.92</td>
<td>84.84</td>
<td>85.56</td>
<td>80.87</td>
<td>84.62</td>
<td>0.001</td>
</tr>
<tr>
<td>BNB</td>
<td>75.45</td>
<td>79.42</td>
<td>71.48</td>
<td>74.73</td>
<td>74.37</td>
<td>75.09</td>
<td>0.002</td>
</tr>
<tr>
<td>BJ4</td>
<td>84.48</td>
<td>89.53</td>
<td>88.45</td>
<td>87.00</td>
<td>85.20</td>
<td>86.93</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 3.5: Percentage classification accuracy given by the 5 × 2-fold cross validation on $D_{acc}$.

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>Avg.</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>86.65</td>
<td>85.37</td>
<td>87.92</td>
<td>87.60</td>
<td>85.21</td>
<td>86.55</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>NB</td>
<td>83.62</td>
<td>74.09</td>
<td>79.97</td>
<td>80.29</td>
<td>79.65</td>
<td>79.52</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>BN</td>
<td>83.31</td>
<td>82.51</td>
<td>83.31</td>
<td>84.90</td>
<td>80.76</td>
<td>82.96</td>
<td>0.001</td>
</tr>
<tr>
<td>ALR</td>
<td>88.24</td>
<td>87.60</td>
<td>87.60</td>
<td>86.65</td>
<td>86.49</td>
<td>87.31</td>
<td>0.002</td>
</tr>
<tr>
<td>J4</td>
<td>84.42</td>
<td>86.17</td>
<td>85.69</td>
<td>86.33</td>
<td>86.32</td>
<td>85.79</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>BNB</td>
<td>83.78</td>
<td>74.88</td>
<td>81.88</td>
<td>79.01</td>
<td>80.45</td>
<td>80.00</td>
<td>0.002</td>
</tr>
<tr>
<td>BJ4</td>
<td>87.28</td>
<td>89.83</td>
<td>88.39</td>
<td>90.78</td>
<td>88.08</td>
<td>88.87</td>
<td>0.001</td>
</tr>
<tr>
<td>MLP</td>
<td>90.61</td>
<td>80.14</td>
<td>87.36</td>
<td>87.73</td>
<td>85.92</td>
<td>86.35</td>
<td>0.028</td>
</tr>
<tr>
<td>NB</td>
<td>78.34</td>
<td>80.14</td>
<td>76.90</td>
<td>78.34</td>
<td>78.34</td>
<td>78.41</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>BN</td>
<td>77.26</td>
<td>83.03</td>
<td>79.42</td>
<td>83.39</td>
<td>81.95</td>
<td>81.01</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>ALR</td>
<td>87.00</td>
<td>85.92</td>
<td>83.03</td>
<td>87.00</td>
<td>84.84</td>
<td>85.56</td>
<td>0.002</td>
</tr>
<tr>
<td>J4</td>
<td>84.48</td>
<td>84.48</td>
<td>78.70</td>
<td>84.84</td>
<td>81.23</td>
<td>82.74</td>
<td>0.002</td>
</tr>
<tr>
<td>BNB</td>
<td>79.42</td>
<td>80.87</td>
<td>79.42</td>
<td>80.51</td>
<td>79.42</td>
<td>79.93</td>
<td>$&lt; 0.001$</td>
</tr>
<tr>
<td>BJ4</td>
<td>84.84</td>
<td>85.92</td>
<td>85.56</td>
<td>84.84</td>
<td>87.00</td>
<td>85.63</td>
<td>$&lt; 0.001$</td>
</tr>
</tbody>
</table>
3.4.2.3 Measuring the Impact of Vital Signs

It is important to highlight that a $5 \times 2$-fold cross validation is not intended to measure the classification accuracy but to rather find differences in the overall accuracy of the classifiers. This is because a two-fold cross validation only uses half of the dataset for training. Now, the actual classification accuracy was measured by a $5 \times 10$-fold cross validation. After evaluating the best classifiers in each dataset for all five random seeds, the overall accuracy for $ALR_{5s}$ was 95.37% whereas $BJ48_{5s}^{acc}$ only reached 90.02%. This is a reduction of 53% in the error rate. A more detailed analysis was carried out for each class (i.e., activity) by calculating a number of performance metrics: precision, recall, F-measure, false positive rate (FPR), and false negative rate (FNR) — their definitions were presented in Section 2.2.2.4. Figure 3.7 compiles the results of the evaluation metrics. Observe that all of them confirm that physiological signals are useful to enhance the recognition of all activities. The most noticeable improvements are for ascending, descending, and walking. Indeed, the FPR was reduced in 52% for walking, 67% for ascending, and 72% for descending. At the same time, the FNR was reduced in 69% for walking and nearly 50% for ascending and descending. This was expected as acceleration signals tend to be similar for ascending, walking, and descending whereas vital signs provide more clear patterns to distinguish among these activities (see Figure 3.1).

3.4.2.4 Analyzing the Impact of Transient Features

To measure the effectiveness of transient features, two new datasets were evaluated: $D_{acc+str}$ and $D_{acc+tra}$. The former incorporates statistical features from acceleration signals along with structural features from physiological signals. The latter includes statistical features from acceleration signals, transient features, and the parameter $b$ of the line $y(t) = mt + b$ that best fits the points of the signal. This last feature is included in $D_{acc+tra}$ to describe, not only the shape, but also the values in the signal. In this manner, a heart rate signal that linearly varies from 80 bpm to 70 bpm, for instance, could be differentiated
Figure 3.7: Impact of physiological signals in the classification accuracy: precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure.
Table 3.6: Impact of physiological signals in the classification accuracy: precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure.

(A)

<table>
<thead>
<tr>
<th>Overall accuracy: 95.37%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ALR_{vs}^{ss}$</td>
</tr>
<tr>
<td>Running</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>Recall</td>
</tr>
<tr>
<td>F-measure</td>
</tr>
<tr>
<td>FPR</td>
</tr>
<tr>
<td>FNR</td>
</tr>
</tbody>
</table>

(B)

<table>
<thead>
<tr>
<th>Overall accuracy: 90.03%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BJ4S_{acc}^{ss}$</td>
</tr>
<tr>
<td>Running</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>Recall</td>
</tr>
<tr>
<td>F-measure</td>
</tr>
<tr>
<td>FPR</td>
</tr>
<tr>
<td>FNR</td>
</tr>
</tbody>
</table>

from another one which changes from 110 bpm to 100 bpm, given that they might represent different activities.

Tables 3.7-(A) and 3.7-(B) display the overall accuracy for $D_{acc+str}$ and $D_{acc+tra}$, respectively. Notice that the accuracies for $D_{vs}$, $D_{acc+str}$, and $D_{acc+tra}$ are very similar, being slightly higher for $D_{acc+tra}$. This implies that transient features could substitute the coefficients of the polynomials that best fit the physiological signals, thereby simplifying the model and reducing the computational complexity of the feature extraction process. This is since there are only two transient features per attribute (i.e., the signed magnitude of change and the intersection with the Y axis) versus nine polynomial coefficients.

The last experiment was comparing transient features to linear regression features. Therefore, the last dataset evaluated was $D_{acc+lr}$, which contains statistical features from acceleration signals and linear coefficients (i.e., slope and intersection with the Y-axis) from physiological signals. The results for $D_{acc+lr}$ are included in Table 3.7 (C). Notice that the
Table 3.7: Impact of transient features in the classification accuracy: precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure.

### (A) Statistical and structural features \((D_{acc+str})\)

<table>
<thead>
<tr>
<th>Overall accuracy: 95.48%</th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
<th>Ascending</th>
<th>Descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.995</td>
<td>0.944</td>
<td>0.960</td>
<td>0.925</td>
<td>0.956</td>
</tr>
<tr>
<td>Recall</td>
<td>0.951</td>
<td>0.974</td>
<td>0.986</td>
<td>0.860</td>
<td>0.927</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.973</td>
<td>0.959</td>
<td>0.973</td>
<td>0.891</td>
<td>0.941</td>
</tr>
<tr>
<td>FPR</td>
<td>0.001</td>
<td>0.039</td>
<td>0.011</td>
<td>0.008</td>
<td>0.006</td>
</tr>
<tr>
<td>FNR</td>
<td>0.049</td>
<td>0.026</td>
<td>0.014</td>
<td>0.140</td>
<td>0.073</td>
</tr>
</tbody>
</table>

### (B) Statistical and transient features \((D_{acc+tra})\)

<table>
<thead>
<tr>
<th>Overall accuracy: 95.654%</th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
<th>Ascending</th>
<th>Descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.978</td>
<td>0.942</td>
<td>0.967</td>
<td>0.950</td>
<td>0.967</td>
</tr>
<tr>
<td>Recall</td>
<td>0.952</td>
<td>0.982</td>
<td>0.988</td>
<td>0.884</td>
<td>0.889</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.965</td>
<td>0.962</td>
<td>0.978</td>
<td>0.916</td>
<td>0.926</td>
</tr>
<tr>
<td>FPR</td>
<td>0.004</td>
<td>0.041</td>
<td>0.009</td>
<td>0.006</td>
<td>0.004</td>
</tr>
<tr>
<td>FNR</td>
<td>0.048</td>
<td>0.018</td>
<td>0.012</td>
<td>0.116</td>
<td>0.111</td>
</tr>
</tbody>
</table>

### (C) Statistical and linear regression \((D_{acc+lr})\)

<table>
<thead>
<tr>
<th>Overall accuracy: 95.27%</th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
<th>Ascending</th>
<th>Descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.985</td>
<td>0.949</td>
<td>0.957</td>
<td>0.906</td>
<td>0.956</td>
</tr>
<tr>
<td>Recall</td>
<td>0.947</td>
<td>0.975</td>
<td>0.984</td>
<td>0.887</td>
<td>0.893</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.965</td>
<td>0.962</td>
<td>0.970</td>
<td>0.896</td>
<td>0.924</td>
</tr>
<tr>
<td>FPR</td>
<td>0.003</td>
<td>0.035</td>
<td>0.011</td>
<td>0.011</td>
<td>0.006</td>
</tr>
<tr>
<td>FNR</td>
<td>0.053</td>
<td>0.025</td>
<td>0.016</td>
<td>0.113</td>
<td>0.107</td>
</tr>
</tbody>
</table>
Table 3.8: Confusion matrix for the best classifier $ALR_{acc+tra}^{5s}$ after five iterations with different random seeds.

<table>
<thead>
<tr>
<th></th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
<th>Ascending</th>
<th>Descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>499</td>
<td>12</td>
<td>12</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Walking</td>
<td>5</td>
<td>1236</td>
<td>7</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Still</td>
<td>4</td>
<td>4</td>
<td>636</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Ascending</td>
<td>2</td>
<td>33</td>
<td>4</td>
<td>298</td>
<td>3</td>
</tr>
<tr>
<td>Descending</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>12</td>
<td>336</td>
</tr>
</tbody>
</table>

computational complexity for calculating the features in $D_{acc+lr}$ and $D_{acc+tra}$ is the same, but the accuracy for the latter is slightly higher.

### 3.4.3 Confusion Matrix

The confusion matrix for the best classifier, $ALR_{acc+tra}^{5s}$, is shown in Table 3.8 after five iterations using five different random seeds. Confusions are, on average, less than 5%, and mostly among three activities: walking, ascending, and descending. This was expected since these three activities exhibit similar patterns depending on the intensity at which they are performed by the individual.

### 3.5 Concluding Remarks

In this chapter, physiological signals have been shown to be an important source of information to improve the recognition of human activities. As a matter of fact, incorporating vital signs—besides the traditionally used acceleration signals—accounts for a reduction of 53% in the error rate. Such improvement is especially significant for ascending, descending, walking, activities that could be confusing from the acceleration point of view. The overall accuracy was up to 95.65%.

Of course, in order to measure physiological signals, additional sensors and wireless communication are required, introducing higher energy expenditures and obtrusiveness. All these aspects should be carefully evaluated before extrapolating a HAR system to a real world application. Finally, transient features were introduced as an efficient alternative to
describe underlying patterns within physiological signals. They can be computed in linear
time and are an important piece to achieve the highest recognition accuracy.
Chapter 4: Mobile Activity Recognition in Real Time

4.1 Note to the Reader

Part of this work was published in [34], © 2012 IEEE. Reprinted, with permission, from “O. D. Lara and M. A. Labrador, A mobile human activity recognition system, in IEEE Consumer Communications and Networking Conference (CCNC), jan. 2012, pp. 38-39”. Appendix A includes the permissions to reuse such work in this dissertation. The corresponding article may be found in Appendix C.

4.2 Introduction

Chapter 3 showed that human activities can be effectively recognized with acceleration and physiological signals. However, such recognition and the performance analyses were carried out offline, in a server. This chapter focuses on the next step in HAR: the online implementation (i.e., providing real-time feedback) on mobile devices. Such task is not trivial in view of the energy, memory, and computational constraints present in the devices. In activity recognition applications, those limitations are particularly critical since they require data preprocessing, feature extraction, classification, and transmission of large amounts of raw data. Furthermore, to the best of our knowledge, available machine learning API’s such as Weka [30] and JDM [31] are neither optimized nor fully functional under current mobile platforms. This fact accentuates the necessity for an efficient mobile library to evaluate machine learning algorithms and implementing HAR systems in mobile devices.

A mobile HAR system will bring important advantages and benefits. In the first place, energy expenditures are expected to be substantially reduced as raw data would not have to be continuously sent to a server for processing. The system would also become more
robust and responsive because it would not depend on unreliable wireless communication links, which may be unavailable or error prone. Finally, a mobile HAR system would be more scalable since the server load would be alleviated by the locally performed feature extraction and classification computations.

This chapter introduces Vigilante, a mobile framework in support of real-time human activity recognition under the Android platform—reported as best-selling smartphone platform in 2010 by Canalys [81]. The implementation and evaluation of Vigilante present the following main results and contributions:

- A library for the *mobile evaluation of classification algorithms* (MECLA) was designed and successfully implemented.

- The Weka API was partially integrated in the Android platform to enable the evaluation of a number of classification algorithms for activity recognition.

- An application for real-time HAR was implemented in an Android cellular phone. It uses MECLA as well as Weka, and it supports multiple sensing devices integrated in a Body Area Network (BAN).

- The evaluation shows that the system can be effectively deployed on current cellular phones in three main regards:

  - *Accuracy*: Human activities are recognized with an overall accuracy of up to 96.8%.

  - *Response time*: The total computational time required for preprocessing, feature extraction, and classification accounts for less than 4% of the window length.

  - *Energy consumption*: The application is able to run for up to 12.5 continuous hours and it enables energy savings of up to 27% with respect to a system that sends all the raw data to the server for remote processing.
4.3 Existing Mobile Real-time HAR Systems

Human activity recognition is a well studied field yet very few works have successfully been deployed in mobile phones. In 2010, Berchtold et al. introduced ActiServ as an activity recognition service for mobile phones [29]. They make use of a fuzzy inference system to classify daily activities, achieving up to 97% of accuracy. Nevertheless, it requires a runtime duration in the order of days! When their algorithms are executed to meet a feasible response time, the accuracy drops to 71%. ActiServ can also reach up to 90% after personalization, in other words, a subject-dependent analysis (i.e., the system needs to be re-trained for new users). Brezmes et al. [28] proposed a mobile application for HAR under the Nokia platform; but they used the $k$-nearest neighbors classifier, which is not scalable for mobile phones as it would need the entire training set—which can be fairly large—to be stored in the device. Finally, Riboni et al. [27] presented COSAR, a framework for context-aware activity recognition using statistical and ontological reasoning under the Android platform. The system recognizes a number of activities very accurately yet it depends on the GPS sensor and an additional accelerometer on the user’s wrist. This introduces higher energy expenditures and privacy concerns. Furthermore, their proposed historical variant increases the response time of the system.

4.4 The Proposed System

Figure 4.1 illustrates the main components of Vigilante and their interrelationship. Three wearable sensors have been integrated: the phone accelerometer, the GPS, and the Bioharness BT sensor strap, manufactured by Zephyr technology [80]. The measured attributes are collected by the mobile application, which executes the data preprocessing, feature extraction, activity recognition, and visualization modules. The application server performs several tasks. In the first place, it delivers previously trained classification models that can be downloaded by the mobile application. In this fashion, improved classification algorithms, and new feature extraction methods can be easily available on the phone. In
the second place, the application server provides authentication, session management, and records additional training data according to the user’s preferences. Finally, the database server runs a PostgreSQL database to store sessions, user information, training data, classification models, among other miscellaneous data.

The rest of this chapter is devoted to the mobile application; Figure 4.2 shows its design: first, the communication and sensing modules allow for collecting raw data from the sensing devices. These data are decoded and organized in time windows by the data preprocessing component. Then, the feature extraction module extracts statistical and structural features from each time window, producing a feature set instance which is later evaluated by the classification module. The output of the classifier is indeed the recognized activity, displayed by the visualization module and sent to the server for further analysis and historical querying. When the system is working on training mode, all raw data are also sent to the server.
Figure 4.2: Mobile application design (© 2012 IEEE [34]).
4.4.1 Sensing Devices

Vigilante currently supports three sensing devices, namely the phone GPS, phone accelerometer, and the BioHarness™ BT chest sensor strap, manufactured by Zephyr. The strap is unobtrusive, lightweight, and can be easily worn by any person. More information on this sensor can be found in Section 3.3.1.1 and in [80]. Although the phone’s accelerometer and the GPS sensors are available in the Vigilante platform, they were not used for activity recognition purposes.

4.4.2 Communication

The communication module encompasses three levels: (1) receiving raw data from the sensors (via Bluetooth), (2) sending raw data or activity results to the server (via TCP/IP), and (3) querying the database (via HTTP servlets). In the first level, three different types of packets are received from the sensing device: acceleration, vital signs, and electrocardiogram. In the second level, these packets are aggregated and sent to the application server, which decodes and stores them into the database. Finally, in the third level of communication, HTTP servlets allow for validating user credentials and querying the list of activities to be recognized, the list of features to be extracted, as well as the classification models to be used.

4.4.3 Sensing and Data Preprocessing

The sensing component manages and synchronizes the flow of data from all sensing devices, i.e., accelerometer, GPS, and the Bioharness BT strap. Figure 4.3 illustrates the interrelationship among the classes in this module. Sensors are represented as entities that extend from the abstract class Sensor and implement methods to connect, read data, and finalize. Additionally, they periodically report measurements to the SensorManager class through the Observer-Observable pattern [100]. With this model, new sensors can be easily incorporated to the system and they are able to work concurrently—in a different thread—as the class Sensor also implements the Java’s Runnable interface.
Every piece of raw data is represented as a packet, which might contain one or several samples, according to the sensor specifications. Each specific type of packet extends from the abstract `Packet` class, and implements a particular `decode` method. The `SensorManager` class receives all packets and controls the data flow to achieve the recognition of activities.

### 4.4.4 Feature Extraction and Selection

This component provides an efficient implementation of statistical, structural, and transient feature extraction methods for nine attributes: heart rate, respiration rate, breath amplitude, skin temperature, posture, ECG amplitude, and tri-axial acceleration. The methods were presented in Section 3.3.2, accounting for a total of 84 features. Given the mobile devices’ computational constraints, only the most relevant features have been chosen for real-time activity recognition. This process, denominated *feature selection* is very useful in any machine learning context to simplify the model, eliminating redundant features which could even diminish the classification performance. The *correlation based feature se-
Table 4.1: Selected features for mobile activity recognition.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>corr&lt;sub&gt;x,y&lt;/sub&gt;</td>
<td>Correlation among acceleration signal in X and acceleration signal in Y axis</td>
</tr>
<tr>
<td>corr&lt;sub&gt;x,z&lt;/sub&gt;</td>
<td>Correlation among acceleration signal in X and acceleration signal in the Z axis</td>
</tr>
<tr>
<td>MAD&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Mean absolute deviation of the acceleration signal in the X axis</td>
</tr>
<tr>
<td>MAD&lt;sub&gt;y&lt;/sub&gt;</td>
<td>Mean absolute deviation of the acceleration signal in the Y axis</td>
</tr>
<tr>
<td>RMS&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Root Mean Square of the acceleration signal in the X axis</td>
</tr>
<tr>
<td>RMS&lt;sub&gt;y&lt;/sub&gt;</td>
<td>Root Mean Square of the acceleration signal in the Y axis</td>
</tr>
<tr>
<td>σ&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Standard deviation of the acceleration signal in the X axis</td>
</tr>
<tr>
<td>σ&lt;sup&gt;2&lt;/sup&gt;&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Variance of the acceleration signal in the X axis</td>
</tr>
<tr>
<td>b&lt;sub&gt;HR&lt;/sub&gt;</td>
<td>Parameter b of the line y = mt + b that best fits the heart rate signal</td>
</tr>
<tr>
<td>b&lt;sub&gt;ECG&lt;/sub&gt;</td>
<td>Parameter b of the line y = mt + b that best fits the ECG amplitude signal</td>
</tr>
<tr>
<td>η&lt;sub&gt;RR&lt;/sub&gt;</td>
<td>Signed Magnitude of Change of the respiration rate signal</td>
</tr>
</tbody>
</table>

The selected features are listed in Table 4.1. Further methods such as Principal Component Analysis and Genetic Search Feature Selection [30] were also evaluated but they considerably affected the overall classification accuracy.

### 4.4.5 Classification

This module handles the evaluation of previously trained classification algorithms. Two different libraries, Weka and MECLA, were integrated to Vigilante.

#### 4.4.5.1 Towards Mobile Weka

The Waikato Environment for Knowledge Analysis (Weka) is one of the most widely used research tools for machine learning. And, one of its most noticeable features is its Java API, which enables the integration of new classification and evaluation methodologies using the underlying core libraries. As part of this dissertation, the Weka API has been partially integrated to the Android platform. However, it was found that some functionalities are still not available on the mobile phone.

Given that the training phase is generally more expensive than the evaluation, the classifiers used in this work were previously trained on the server. Now, to make them available on the phone, Vigilante uses the Java’s object serialization API, which permits to convert a given class instance (i.e., an object) to a stream of bytes and vice versa. The object
that is intended to be serialized should implement the `java.io.Serializable` interface. This is not an issue since the Weka classifiers extend from the superclass `weka.classifiers.Classifier` which, in turn, implements the `java.io.Serializable` interface. The class `ObjectOutputStream` was used to export each trained classifier object as a binary file. Such file is copied to the phone’s persistent memory and then, by means of the class `ObjectInputStream`, the trained classifier instance is reconstructed from the stream of bytes.

There are, nevertheless, various disadvantages of running Weka in the mobile phone. In the first place, not all the classification methods are supported by the mobile platform. Moreover, the file `weka.jar` should be part of the Android application, increasing the size of the executable APK file in more than 6 MB. This is because Weka libraries comprehend training algorithms and other functionalities that are not necessary for HAR purposes. This fact may also hinder a final application to be massively deployed.

### 4.4.6 MECLA

Given that Weka is not designed nor optimized for existing mobile platforms, a new library, MECLA, is proposed to evaluate classification models on the phone. The current implementation supports decision trees —as a proof of concept—, thereby enabling the C4.5, ID3, Decision Stump, Random Tree, and M5, among other classification algorithms.

The design of the classification module is shown in Figure 4.4. Nodes correspond to features (e.g., mean acceleration in X) whereas edges define relations of an attribute with a numeric or nominal value (e.g., mean acceleration in X is greater than 2.382 g). The method `evaluate` of the class `Edge` returns whether or not a given value fulfills the edge’s relational condition. This method is utilized by class `Node`’s `evaluate` method, which returns the next child node to be visited given a particular attribute value. In this way, the evaluation of a feature vector (i.e., an instance of the class `FeatureSet`) starts at the root node and follows the corresponding branches according to the output of the node evaluation methods. The process stops when the current node does not have any children, and the associated activity class (e.g., running, walking, etc.) is returned.
Figure 4.4: Simplified UML class diagram for the classification component (© 2012 IEEE [34]).
The classification model was priorly trained in Weka using the C4.5 algorithm. The alphanumeric representations of all decision trees, as given by the Weka output, were stored in the database. An HTTP servlet allows the mobile application to query a decision tree model in accordance to the parameter values set by the user. An iterative algorithm was implemented to build a DecisionTree object (i.e., nodes and edges) based upon the alphanumeric representation retrieved by the servlet, in order to generate predictions on the user’s activity.

4.4.7 User Interface

Vigilante has three different user profiles: trainer, tester, and administrator. A tester is a regular user whose activities are to be monitored and recognized. A trainer, as indicated by its name, intended to collect training data that can be used to build additional classification models. In this profile, the user is also required to enter the real performed activity as a ground truth. Finally, the administrator is able to do training, testing, and modifying the application settings.

Figure 4.5 displays the two main screens of Vigilante, namely the monitor and the configuration. The former displays the sensed data (e.g., heart rate, respiration rate, skin temperature, etc.), the current user’s activity, and the data collection time in real time. The latter, allows the administrator to set parameters such as the window size, number of aggregated packets, maximum number of Bluetooth reconnections, time between Bluetooth reconnections, and feature extraction methods. It also permits turning on and off each individual sensor, as well as enabling or disabling data transmission to the server.

4.5 Evaluation

4.5.1 Experiment Design

Vigilante was tested on an HTC Evo 4G mobile phone, being compatible with Android 2.1 or higher. Four individuals: $I_1$ through $I_4$—three males and a female—performed each activity in a sequential fashion during two to five minutes. $I_1$ was part of the training data.
Figure 4.5: Mobile application user interface (© 2012 IEEE [34]).

Table 4.2: Physical characteristics of the new participants.

<table>
<thead>
<tr>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (years)</td>
</tr>
<tr>
<td>Weight (kg)</td>
</tr>
<tr>
<td>Height (m)</td>
</tr>
<tr>
<td>BMI (kg/m²)</td>
</tr>
</tbody>
</table>

collection phase, while individuals I₂ through I₄ are new to the system. The physical characteristics of the new individuals are summarized in Table 4.2. A naturalistic experiment was carried out, i.e., no instructions were given to the participants on how to perform the activities.

Table 4.3 includes the most important parameters for all the experiments. The following classification methods were attempted on the phone using the Weka libraries:

- \textit{J₄₈}: the C4.5 decision tree.
- \textit{BayesNet}: a Bayesian Network classifier using the K2 search algorithm.
- \textit{SMO}: the Sequential Minimal Optimization algorithm.
Table 4.3: Parameters to evaluate Weka classifiers in mobile devices.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>HTC Evo 4G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Android 2.X</td>
</tr>
<tr>
<td>Dataset</td>
<td>$D_{os}$</td>
</tr>
<tr>
<td>Features</td>
<td>12</td>
</tr>
<tr>
<td>Instances</td>
<td>629</td>
</tr>
<tr>
<td>Overlapping</td>
<td>50%</td>
</tr>
</tbody>
</table>

- **IBK**: the k-nearest neighbors classifier.
- **MultilayerPerceptron**: a neural network using backpropagation.
- **Logitboost**: The Additive Logistic Regression classifier.
- **Bagging**: an ensemble of ten $J_{48}$ base learners using the bootstrap aggregation technique.

It was found, nonetheless, that some of the Weka’s classifier implementations are not suitable for mobile devices. Indeed, the $J_{48}$ (i.e., the C4.5 decision tree), Bagging, and the MultilayerPerceptron generated stack overflow exceptions in the process of de-serialization.

The reader may recall that Chapter 3 suggests ALR as the most appropriate classifier for the HAR problem addressed in this dissertation. Since that algorithm was supported by the Android platform, it was used for the experiments with the Weka API. On the other hand, the C4.5 algorithm was evaluated using the MECLA libraries. This is with the aim of comparing the performance of Weka and MECLA. The evaluation encompasses three main aspects: accuracy, response time, and energy consumption.

### 4.5.2 Accuracy

It is worth to highlight that the evaluation presented in this dissertation was accomplished online. Instead, previous studies which implemented real-time activity recognition calculate the confusion matrices and other performance metrics offline [28, 27], applying pre-processing and filtering the data. These two stages are helpful to remove noise but require...
human intervention, which makes them unavailable offline. Thus, the results presented in this section are more realistic.

To avoid potential mistakes when labeling the data, the assessment of the classification accuracy was done programatically. In training mode, each user set the real activity as the ground truth, and performed said activity for certain amount of time. Meanwhile, the application computes the classified activity for each time window using the ALR classifier powered by Weka, as well as the MECLA’s C4.5 classifier implementation. Given both the predicted and the real activity, the confusion matrices are directly calculated and displayed on the phone. The results for individual $I_1$ are shown in Table 4.4, where columns correspond to the real activity and rows represent the classified activity. The overall accuracy was very similar for both classifiers, i.e., 96.39% for ALR and 96.84% for C4.5, demonstrating the effectiveness of Weka and MECLA. This seems to contradict the results of Chapter 3, which suggest that ALR is significantly more accurate than J48. Nevertheless, that study uses cross validation among all the user’s data to measure the overall accuracy. In this case a user-specific analysis was done —because individual $I_1$ also participated in the training phase—, which favors higher accuracy levels for activity recognition [20].
Table 4.5: Confusion matrix for Individuals 2 and 3 (5 activities).

### ALR (WEKA). Overall accuracy: 69.38%

<table>
<thead>
<tr>
<th></th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
<th>Ascending</th>
<th>Descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>53</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Walking</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>20</td>
<td>72</td>
</tr>
<tr>
<td>Still</td>
<td>0</td>
<td>0</td>
<td>124</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ascending</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>48</td>
<td>18</td>
</tr>
<tr>
<td>Descending</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>58</td>
</tr>
</tbody>
</table>

### C4.5 (MECLA). Overall accuracy: 64.05%

<table>
<thead>
<tr>
<th></th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
<th>Ascending</th>
<th>Descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>70</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Walking</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>1</td>
<td>79</td>
</tr>
<tr>
<td>Still</td>
<td>0</td>
<td>0</td>
<td>124</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ascending</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>11</td>
<td>56</td>
</tr>
<tr>
<td>Descending</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>54</td>
</tr>
</tbody>
</table>

A second experiment considers individuals $I_2$ and $I_3$, which did not participate in the training collection phase. Tables 4.5-A and 4.5-B display the confusion matrices for both classifiers. Notice that here the overall accuracy is quite lower than for $I_1$ (between 64% and 69%). This was expected as the gait and the intensity of the activities are individual-specific, specially for ascending and descending. Another important factor is that $I_2$ and $I_3$ have different physical characteristics than the individuals who collected training data for ascending and descending. In Chapter 6, the hypothesis of a group-specific data collection methodology is presented to overcome this issue.

A last experiment was accomplished with individuals $I_2$ and $I_4$ but only considering three activities: walking, running, and still. The classification model generated by the C4.5 algorithm is shown in Figure 4.6. Four features are part of the tree: (1) $MAD(AccX)$, i.e., the maximum absolute deviation of the acceleration in the X axis; (2) $RMS(AccX)$, i.e., the root mean square of the acceleration in the X axis, (3) $MoC(HR)$, i.e., the magnitude of change of the heart rate signal, and (4) $Slope(Temp)$, i.e., the slope of the line that best fits the skin temperature signal. In this case, even though these two individuals did not participate in the training phase, the system’s accuracy was more than acceptable (92.25%). This brings a very interesting point of discussion: some activities are user-specific, while
Table 4.6: Confusion matrix for Individuals 2 and 4 with three activities (© 2012 IEEE [34]).

<table>
<thead>
<tr>
<th></th>
<th>Running</th>
<th>Walking</th>
<th>Still</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>123</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Walking</td>
<td>5</td>
<td>112</td>
<td>21</td>
</tr>
<tr>
<td>Still</td>
<td>0</td>
<td>1</td>
<td>105</td>
</tr>
</tbody>
</table>

Overall accuracy: 92.25%

others can be generalized for individuals with different physical characteristics. Therefore, to improve the recognition accuracy in this particular case study, the system would only need to be re-trained for two activities: ascending and descending.

Figure 4.6: Classification model generated by the C4.5 algorithm with three activities (© 2012 IEEE [34]).

4.5.3 Response Time

The response time was measured for each time window as the total time spent by (1) preprocessing, (2) feature extraction, and (3) classification. The total average response time after issuing 100 time windows was about 171 ms. The most demanding stage was the preprocessing phase, which, on average, consumed 53% of the total response time. Feature extraction consumed about 45% of the time and classification only required roughly 2% of
the total computation time. This was expected as the feature extraction algorithms run in $O(n)$, for 250 samples. On the other hand, the classification runs in $O(\log m)$ for the decision tree —for $m$ nodes— while it requires constant time for the ALR classifier —since it evaluates ten one-level decision trees and then uses voting. Overall, the response time is less than 4% of the window length, confirming that Vigilante can be successfully deployed in current cellular phones.

4.5.4 Energy Consumption

One of the hypotheses of this chapter is that executing activity recognition locally in the phone —rather than sending all raw data to the server— is effective to save energy. In order to prove it, three cases were considered: (1) recognizing activities locally in the phone without sending raw data to the server; (2) sending all raw data to the server to recognize the activities remotely; and (3) not running the application to measure the energy consumed by the operating system and other phone services alone. The classes `Intent` and `BatteryManager` of the Android API were used to measure the battery charge difference after three continuous hours of use. Then, energy consumption was estimated given that the phone’s battery works at 3.7 V and has a total charge of 1500 mAh. The results, shown in Table 4.7, indicate that performing local activity recognition allows for increasing the application lifetime in 25%. More precisely, the system can run for up to 12.5 hours in case 1, versus 9.38 hours in case 2. Now, excluding the energy consumed by the operating system, the total energy savings in case 1 with respect to case 2 were roughly 26.7%. This result becomes remarkable as case 2 was already using data aggregation as an energy saving mechanism.

Table 4.7: Estimated energy consumption after executing the application for three hours. (© 2012 IEEE [34]).

<table>
<thead>
<tr>
<th>Case</th>
<th>Charge diff.</th>
<th>Current</th>
<th>Power</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.36 Ah (24%)</td>
<td>0.12 A</td>
<td>0.444 W</td>
<td>4795.2 J</td>
</tr>
<tr>
<td>2</td>
<td>0.48 Ah (32%)</td>
<td>0.16 A</td>
<td>0.592 W</td>
<td>6396.6 J</td>
</tr>
<tr>
<td>3</td>
<td>0.03 Ah (2%)</td>
<td>0.01 A</td>
<td>0.037 W</td>
<td>399.6 J</td>
</tr>
</tbody>
</table>
4.6 Concluding Remarks

This chapter presented Vigilante as a mobile framework for real-time human activity recognition under the Android platform. The system features a library for mobile evaluation of classifiers (MECLA), which can be utilized in further machine learning applications as an alternative to Weka. MECLA becomes useful since it enables decision-tree based algorithms which are unavailable using Weka libraries. The evaluation shows that current cellular phones are more than capable to run Vigilante. Besides, a substantial reduction in energy consumption was found (up to 26.7%) compared to a server-based HAR system.
In a machine learning context, the integration of a committee of experts (i.e., classification models) is shown to be beneficial to improve the classification accuracy of an individual model [87]. Since classification is one of the most important components in activity recognition, this chapter explores new strategies to combine a set of learners in a multiple classifier system (MCS). Particularly, the sort of MCS studied in this chapter is intended to solve the problem stated in Definition 3 (see Chapter 1). That definition assumes the following input is provided:

- A classification problem with a set $\Omega = \{\omega_0, \ldots, \omega_{n-1}\}$ of $n$ classes
- A dataset with $N$ instances
- A set $D = \{D_0, \ldots, D_{k-1}\}$ of classifiers
- A set $S = \{s_0, \ldots, s_{k-1}\}$ with $k$ predictions from each classifier for a given instance $x$, such that $s_i = D_i(x)$
- A set $M = \{M^0, \ldots, M^{k-1}\}$ of $k$ confusion matrices for each classifier, where the sum of columns is the total number of predictions for each class and the sum of rows is the number of actual instances.

The goal is to find the correct label $\omega^*$ iff $\exists s_i \in S$ such that $\omega^* = s_i$. An intuitive solution to this problem is to estimate the probability that each prediction $s_i \in S$ is correct and then select the prediction $s^*$ with the maximum probability. Another approach is to rather estimate the probability of each class $\omega_i$ by combining the predictions of all classifiers. Nonetheless, selecting the prediction or the class with the highest estimated probability is...
not always an appropriate solution, especially when some classes are harder to distinguish than others. If we are dealing with recognizing physical activities, for instance, sitting and running are clearly differentiable, so higher probabilities are expected for these classes. Instead, walking upstairs and walking downstairs might be confusing in some cases [33], yielding to smaller probabilities. Therefore, always choosing the prediction with the highest probability value may bias the ensemble to select the easiest classes, thereby affecting the overall accuracy and diversity.

This dissertation aims to address aforementioned issues by introducing two new probabilistic strategies for fusion and selection in a multiclassifier system. Both of them are based on simple heuristic rules, maintaining ease of implementation and low computational cost. An extensive analysis with seven classification algorithms and eleven datasets demonstrates that the proposed methods are effective to improve classification accuracy with respect to the individual classifiers and other well known fusion and selection algorithms. Also, an algorithm to select base classifiers is proposed in order to guarantee a significant accuracy improvement by the proposed strategies.

5.1 Probabilistic Strategies in Multiple Classifier System

5.1.1 Failure Product

As seen in Section 2.4, the probability $\hat{P}(\omega^* = \omega_j \mid s_i = \omega_j)$ that a prediction $s_i = D_i(x)$, corresponding to class $\omega_j$, is correct can be expressed as follows:

$$\hat{P}(\omega^* = \omega_j \mid s_i = \omega_j) = \frac{M^j_{\omega_j}}{\sum_{r=0}^{n-1} M^r_{\omega_j}} \quad (5.1)$$

Given this result, it would sound logical to always select the prediction $s^+$ with the highest probability, as proposed by Giacinto et al. [93]:

$$s^+ = \arg \max_{s_i \in S} \left\{ \hat{P}(\omega^* = \omega_j \mid s_i = \omega_j) \right\} \quad (5.2)$$
Nonetheless, this estimation is highly biased when some classes are harder to predict than others, affecting the diversity of classification and, of course, the overall accuracy. This approach may also make an incorrect decision if several classifiers support a prediction $s_r$ different than $s^+$. Consider the case in Figure 5.1. Classifier $D_0$ predicted class $\omega_1$ with the highest probability (i.e., 0.9). But all other classifiers selected $\omega_2$ with high probability (i.e., among 0.8 and 0.85). In this scenario, prediction $s_0 = \omega_1$ is very likely to be wrong although it has the maximum probability.

Table 5.1: Highest probability fallacy

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0 = \omega_1$</td>
<td>0.9</td>
</tr>
<tr>
<td>$s_1 = \omega_2$</td>
<td>0.85</td>
</tr>
<tr>
<td>$s_2 = \omega_2$</td>
<td>0.82</td>
</tr>
<tr>
<td>$s_3 = \omega_2$</td>
<td>0.8</td>
</tr>
</tbody>
</table>

In such direction, a new metric is proposed. From Equation 5.1, the probability that prediction $s_i$ is not correct, would be given by:

$$\hat{P}(\omega^* \neq \omega_j \mid s_i = \omega_j) = 1 - \frac{M^i_{\omega_j \omega_j}}{\sum_{r=0}^{n-1} M^i_{r \omega_j}}$$  \hspace{1cm} (5.3)

**Definition 9 (Failure product)** The failure product $FP(\omega_j)$ for a given class $\omega_j$ is defined as follows:

$$FP(\omega_j) = \left\{ \begin{array}{ll} \infty & \text{if } s_i \neq \omega_j \forall s_i \in S \\ \prod_{i|s_i=\omega_j} \left[ 1 - \frac{M^i_{\omega_j \omega_j}}{\sum_{r=0}^{n-1} M^i_{r \omega_j}} \right] & \text{otherwise} \end{array} \right. \hspace{1cm} (5.4)$$

Then, the prediction $s^*_{FP}$ with the smallest failure product will be the output of the multiple classifier system:
\[
s_{FP}^* = \arg \min_{\omega_j \in \Omega} \{FP(\omega_j)\} \tag{5.5}
\]

### 5.1.2 Precision-recall Difference

The precision is a natural estimation for the probability that a prediction \(s_i\) is correct. However, better estimations could be done by also considering the recall of those classifiers which have a different prediction than \(s_i\). Observe the situation in Figure 5.2. For a given instance \(x\), the prediction \(s_0\) by classifier \(D_0\) is \(\omega_1\) whereas the output \(s_1\) of \(D_1\) is \(\omega_0\). Notice that the precision for prediction \(s_0\) (i.e., 0.8) is higher than the precision of \(s_1\) (i.e., 0.7). However, \(D_1\) has a very high recall for class \(\omega_1\) (i.e., 0.99) so it is very unlikely that \(D_1\) misses an instance of class \(\omega_1\). Hence, a more informed decision should not ignore \(s_1 = \omega_0\) as a potential prediction.

Table 5.2: Precision vs. recall fallacy

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_0)</td>
<td>(\omega_0)</td>
<td>0.9</td>
<td>0.5</td>
<td>(\omega_0)</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>(\omega_1)</td>
<td><strong>0.8</strong></td>
<td>0.6</td>
<td>(\omega_1)</td>
<td>0.4</td>
<td><strong>0.99</strong></td>
</tr>
<tr>
<td></td>
<td>(\omega_2)</td>
<td>0.4</td>
<td>0.7</td>
<td>(\omega_2)</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The recall of classifier \(D_i\) in a particular class can be estimated from the confusion matrix in the same fashion as the precision:

\[
\text{recall}_i(s_i) = \hat{P}(s_i = \omega_j | \omega^* = \omega_j) = \frac{TP_{ij}}{TP_{ij} + FN_{ij}} = \frac{M_{i\omega_j \omega_j}}{\sum_{r=0}^{n-1} M_{i\omega_j \omega_r}} \tag{5.6}
\]

Now, a metric that incorporates both precision and recall is defined to estimate the quality of each base classifier’s prediction.

**Definition 10 (Precision-recall difference)** The precision-recall difference \(\text{PRD}(s_i)\) for a given prediction \(s_i\) is defined as follows:
\[ PRD(s_i) = \frac{M^i_{\omega_i/\omega_j}}{\sum_{r=0}^{n-1} M^i_{r/\omega_j}} - \max_{t|s_i \neq \omega_j} \left\{ \frac{M^t_{\omega_i/\omega_j}}{\sum_{r=0}^{n-1} M^t_{\omega_j/r}} \right\} \]  \hspace{1cm} (5.7)

The first term of this difference is nothing but the precision of classifier \( D_i \) for class \( s_i = \omega_j \). The second term is the maximum recall of all other classifiers with a prediction different than \( s_i \). The main idea behind this strategy is to penalize predictions that were rejected by classifiers with a high recall. It was experimentally found that the max operator performs better than the average in this case. The value of the precision-recall difference is in the interval \([-1, 1]\) and the prediction \( s^*_{PR} \) with the maximum \( PRD \) will be returned by the MCS:

\[ s^*_{PR} = \arg \max_{0 < i < k-1} \{PRD(s_i)\} \]  \hspace{1cm} (5.8)

5.2 Evaluation

5.2.1 Experiment Design

An experimental analysis was carried out to assess the effectiveness of the proposed strategies. All the ensembles were implemented in Java using the Weka API [30]. Eleven different datasets were evaluated in this work. Most of them are part of the UCI machine learning repository [101]. The datasets and their characteristics are described in Table 5.3.

Seven base classification algorithms were considered, namely Nave Bayes (NB), Bayesian Network (BN), Instance Based Learning (IBK), Repeated Incremental Pruning (RIP), C4.5 decision tree, Logistic Regression (LR), and Sequential Minimum Optimization (SMO).

The two proposed strategies were compared to three well known classifier fusion and selection approaches:

- \textit{Naïve Bayes Combination} (NBC) [87] estimates the probability of each class under the assumption that the classifiers are conditionally independent given a class \( \omega_i \), using
Table 5.3: Dataset specifications.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attributes</th>
<th>Classes</th>
<th>Instances</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance-scale</td>
<td>4</td>
<td>3</td>
<td>625</td>
<td>[101]</td>
</tr>
<tr>
<td>diabetes</td>
<td>8</td>
<td>2</td>
<td>768</td>
<td>[101]</td>
</tr>
<tr>
<td>glass</td>
<td>9</td>
<td>6</td>
<td>135</td>
<td>[101]</td>
</tr>
<tr>
<td>$D_{acc+tra}$</td>
<td>90</td>
<td>5</td>
<td>619</td>
<td>[33]</td>
</tr>
<tr>
<td>$D_{acc}$</td>
<td>90</td>
<td>5</td>
<td>277</td>
<td>[33]</td>
</tr>
<tr>
<td>ionosphere</td>
<td>34</td>
<td>2</td>
<td>351</td>
<td>[101]</td>
</tr>
<tr>
<td>lymph</td>
<td>18</td>
<td>4</td>
<td>148</td>
<td>[101]</td>
</tr>
<tr>
<td>segment</td>
<td>19</td>
<td>7</td>
<td>2310</td>
<td>[101]</td>
</tr>
<tr>
<td>sonar</td>
<td>60</td>
<td>2</td>
<td>208</td>
<td>[101]</td>
</tr>
<tr>
<td>soybean</td>
<td>35</td>
<td>19</td>
<td>683</td>
<td>[101]</td>
</tr>
<tr>
<td>vehicle</td>
<td>18</td>
<td>4</td>
<td>846</td>
<td>[101]</td>
</tr>
</tbody>
</table>

Equation 2.16. The class with the maximum probability is chosen as the ensemble’s output.

- **Plurality** (PL) [92] evaluates all classifiers and selects the prediction with the greatest number of votes.

- **Local Class Accuracy** (LCA) [93] chooses the prediction with the highest precision (see Equation 2.19).

- The **Oracle** (ORA) is the optimal selector and it returns the correct label $s^*$ if $s^* \in S$. Otherwise, it returns a random prediction $s_i$. Evidently, the oracle is not practical to be implemented in real applications as it requires a fully labeled dataset. However, it is shown as an upper bound for the classification accuracy.

Each dataset was divided in three folds: *training set, estimation set, and evaluation set*, as suggested by [102]. As usual, the first one was used to train the base classifiers. Then, the posterior probabilities were estimated from the confusion matrices of each base classifier after using the second fold. Finally, the third fold served to evaluate the performance of both base classifiers and ensembles. The evaluation was repeated three times using each fold as training set and randomly dividing the remaining dataset in two halves (one for
estimation and another for evaluation). This entire process was completed for five different random seeds to provide statistical robustness, and the mean values were calculated.

5.2.2 Results

Table 5.4 displays the percentage average accuracies for each dataset given by all base classifiers and ensembles. The best accuracy values per dataset are in bold, the best ensembles are marked with an asterisk (*), and the best base classifiers are marked with a dot (●).

Note that in eight out of eleven datasets, at least one of the ensembles improved the overall classification accuracy with respect to the base classifiers. In six datasets, the FP strategy achieves the highest accuracy while in four of them, the PRD is the best ensemble. In nine cases, the accuracy of PRD is higher than LCA and PL, whereas in ten cases, FP outperforms LCA and PL. This shows that the proposed strategies are effective to improve classification accuracy with respect to previously proposed fusion and selection methods. However, in three datasets (vehicle, soybean, and balance-scale), none of the ensembles could improve the best base classifier’s performance. It was found that the base classifier set plays a significant role in such matter. This situation is examined next.
5.2.3 Correlation Analysis

A successful multiclassifier system should maintain base classifiers with high diversity. If the base classifiers are rather redundant, the ensemble would not deliver a significant improvement. A well known measure of diversity is the pairwise correlation $\rho_{ij}$ [88] between classifiers $D_i$ and $D_j$, defined as follows:

$$
\rho_{ij} = \frac{ad - bc}{\sqrt{(a + b)(c + d)(a + c)(b + d)}}
$$

(5.9)

where:

- $a$ is the number of instances correctly classified by both classifiers.
- $b$ is the number of instances correctly classified by classifier $D_i$ but incorrectly classified by classifier $D_j$.
- $c$ is the number of instances correctly classified by classifier $D_j$ but incorrectly classified by classifier $D_i$.
- $d$ is the number of instances incorrectly classified by both classifiers.

For each dataset, the correlation matrix $P = [\rho_{ij}]_{k \times k}$ has been calculated. An interesting phenomenon was found for the soybean dataset (see Table 5.5). Observe that all the correlation values are very high (between 0.84 and 0.96). This means that all the classifiers yielded the same predictions for most of the instances, which does not allow for a successful classifier ensemble. Note that even the oracle only improved the classification accuracy by less than 3% in regards to SMO (i.e., the best base classifier). Such analysis suggests that no successful ensemble can be built with the given set of base classifiers. Now, the vehicle and the balance-scale datasets do not follow a clear correlation pattern so a different metric will be utilized to analyze them.
Table 5.5: Correlation matrix for the soybean dataset.

<table>
<thead>
<tr>
<th></th>
<th>NB</th>
<th>BN</th>
<th>RIP</th>
<th>C4.5</th>
<th>LOG</th>
<th>SMO</th>
<th>IBK</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>1</td>
<td>0.96</td>
<td>0.87</td>
<td>0.87</td>
<td>0.88</td>
<td>0.89</td>
<td>0.91</td>
</tr>
<tr>
<td>BN</td>
<td>0.96</td>
<td>1</td>
<td>0.86</td>
<td>0.86</td>
<td>0.88</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>RIP</td>
<td>0.87</td>
<td>0.86</td>
<td>1</td>
<td>0.86</td>
<td>0.84</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>C4.5</td>
<td>0.87</td>
<td>0.86</td>
<td>0.86</td>
<td>1</td>
<td>0.85</td>
<td>0.88</td>
<td>0.86</td>
</tr>
<tr>
<td>LOG</td>
<td>0.88</td>
<td>0.88</td>
<td>0.84</td>
<td>0.85</td>
<td>1</td>
<td>0.91</td>
<td>0.87</td>
</tr>
<tr>
<td>SMO</td>
<td>0.89</td>
<td>0.90</td>
<td>0.86</td>
<td>0.88</td>
<td>0.91</td>
<td>1</td>
<td>0.90</td>
</tr>
<tr>
<td>IBK</td>
<td>0.91</td>
<td>0.90</td>
<td>0.86</td>
<td>0.86</td>
<td>0.87</td>
<td>0.90</td>
<td>1</td>
</tr>
</tbody>
</table>

5.2.4 Collaboration among Classifiers

Although low correlation is a desirable feature, it is not sufficient to guarantee a successful MCS. Imagine, for instance, that an additional classifier which outputs uniformly random predictions is included in the ensemble. That classifier is expected to have a very low correlation with all other classifiers, but its predictions are also expected to be very inaccurate, especially when there are many classes. Of course, such random classifier could affect the overall performance of the ensemble and should be avoided.

Now, in order to properly select the set of the base classifiers, the collaboration matrix $B$ is introduced such that the element $0 \leq b_{ij} \leq 1$ is the proportion of instances correctly classified by classifier $D_i$ but incorrectly classified by classifier $D_j$. Table 5.6 shows the $B$ matrix for the balance-scale dataset; for this dataset, the most accurate classifier was LOG. Note that, in 27% of the cases, the BN classifier disagrees with LOG while the latter is giving correct predictions. At the same time, in only 4% of the cases, the predictions by BN are correct while LOG’s are not. This means that including the BN classifier does not seem to be beneficial but rather detrimental for the ensemble. On the other hand, SMO erroneously disagrees with LOG in only 5% of the instances, but correctly disagrees with LOG in 2% of the cases. Therefore, SMO is more likely to help LOG and contribute to the ensemble performance.

Then, the level of collaboration $-1 \leq \kappa_{ij} \leq 1$ of classifier $D_i$ to classifier $D_j$ is defined as follows:
Table 5.6: $B$ matrix for the balance-scale dataset.

<table>
<thead>
<tr>
<th></th>
<th>NB</th>
<th>BN</th>
<th>RIP</th>
<th>J48</th>
<th>LOG</th>
<th>SMO</th>
<th>IBK</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>0</td>
<td>0.18</td>
<td>0.11</td>
<td>0.12</td>
<td>0.03</td>
<td>0.02</td>
<td>0.10</td>
</tr>
<tr>
<td>BN</td>
<td>0.02</td>
<td>0</td>
<td>0.07</td>
<td>0.07</td>
<td>0.04</td>
<td>0.02</td>
<td>0.06</td>
</tr>
<tr>
<td>JRIP</td>
<td>0.05</td>
<td>0.16</td>
<td>0</td>
<td>0.07</td>
<td>0.05</td>
<td>0.03</td>
<td>0.07</td>
</tr>
<tr>
<td>J48</td>
<td>0.05</td>
<td>0.15</td>
<td>0.07</td>
<td>0</td>
<td>0.05</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td>LOG</td>
<td>0.11</td>
<td>0.27</td>
<td>0.19</td>
<td>0.20</td>
<td>0</td>
<td>0.05</td>
<td>0.17</td>
</tr>
<tr>
<td>SMO</td>
<td>0.06</td>
<td>0.23</td>
<td>0.14</td>
<td>0.15</td>
<td>0.02</td>
<td>0</td>
<td>0.13</td>
</tr>
<tr>
<td>IBK</td>
<td>0.04</td>
<td>0.17</td>
<td>0.09</td>
<td>0.08</td>
<td>0.04</td>
<td>0.03</td>
<td>0</td>
</tr>
</tbody>
</table>

$\kappa_{ij} = b_{ij} - b_{ji}$ \hspace{1cm} (5.10)

Greater values of $\kappa_{ij}$ indicate more chances of $D_j$ being able to help $D_i$. Algorithm 1 formalizes the proposed procedure to select the base classifiers. Initially, the only member of the ensemble is the most accurate classifier (line 3). Then, the classifier $D^+$—which has the greatest average level of collaboration to all $D_j \in E$—is iteratively appended to the set $E \subseteq D$, containing the suggested classifiers to be part of the ensemble.

After executing the LOC selection algorithm to the balance-scale dataset, four classifiers were selected, namely LOG, SMO, NB, and IBK. With this setting, the maximum accuracy was 91.12%, given by the PRD strategy, which is higher than the 90.8% reached by the individual LOG classifier. Figure 5.1 shows the variation of the accuracy versus the number of classifiers using the level of collaboration criterion. Observe that the oracle is always improved when a new classifier is appended to the pool, but this is not the case for the other ensembles. FP reaches its best accuracy for five classifiers, while LCA and PRD achieve their maximum potentials with four classifiers.

Finally, Table 5.7 displays the $B$ matrix for the vehicle dataset. In this case, the best base classifier is also LOG. In 31% of the cases, the LOG classifier predicts the correct class while NB does not, whereas in only 5% of the cases, NB helps LOG. A similar situation occurs for all other classifiers, thus, the level of collaboration with respect to LOG is so low that the ensemble cannot be successful unless different classifiers or different sampling procedures are incorporated.
Algorithm 1: Base classifier selection based on classifier level of collaboration (LOC).

\[ D \equiv \text{the pool of all possible } k \text{ base classifiers of the ensemble} \]
\[ E \equiv \text{the suggested set of base classifiers} \]

1. Compute the \( B_{k \times k} \) matrix for all classifiers in \( D \)
2. \( D^* \leftarrow \text{most accurate classifier in } D \)
3. \( E \leftarrow \{ D^* \} \)
4. \( D \leftarrow D - \{ D^* \} \)
5. while \( D \neq \)
6. \[ D^+ \leftarrow \arg \max_{D_j \in D} \left\{ \frac{1}{\|E\|} \sum_{D_j \in E} \kappa_{ij} \right\} \]
7. if (accuracy of \( E \)) \{ (accuracy of \( E \cup \{ D^+ \} \)) then
8. \( E \leftarrow E \cup \{ D^+ \} \)
9. \( D \leftarrow D - \{ D^+ \} \)
10. else
11. return \( E \)
12. end if
13. end while
14. return \( E \)

5.2.5 MCS in HAR

This chapter has shown that the proposed strategies improve the classification accuracy in a number of machine learning problems. Even for the HAR datasets, the FP and PRD produce —on average— more accurate results than the base classifiers and other ensembles. Nevertheless, even the highest accuracy achieved by FP in HAR (i.e., 93.10%) is not significantly better than for the ALR algorithm presented in Chapter XX (i.e., 93.13% in a 5 \times 2-fold cross validation and up to 95.67% in a 5 \times 10-fold cross validation). Since ALR achieves such a high accuracy, attempting to outperform it could cause overfitting; in other words, noise (i.e., instances with incorrect labels) would become part of the model.
thereby degrading the overall accuracy of the classifier in unseen data. As a matter of fact, one of the issues with the ALR algorithm and other approaches based on boosting is that they poorly tolerate noise [53]. This is because boosting iteratively assigns higher weights to incorrectly classified instances, forcing base classifiers to make such instances part of the model. And, if there is a considerable amount of noise, the classifier will be overfitted.

5.2.5.1 Impact of Noise in HAR

In most machine learning problems, labeling the data requires human intervention, making this process error prone. A more detailed analysis of the impact of noise in pattern classification is presented in [103]. Such work evaluates different types of attribute noise
and class noise with a number of classification algorithms within different paradigms. Particularly, in a human activity recognition context, label errors could occur due to different factors. A malicious or lazy user who is required to follow a routine of exercise could report training data labeled as *running* while he or she is actually *lying* or *sitting*. Also, an incorrect use of the mobile applications presented in Chapters 3 and 4 could lead to mistaken labels if the user selects the wrong activity or they change activities before pressing the *stop* button. These situations emphasize the need for classification algorithms that are able to handle noise in activity recognition.

5.2.5.2 Experiments

In this study, noise was induced by arbitrarily modifying the labels of a subset $Y \in D_{acc+tra}$. The new labels and the instances in $Y$ were chosen uniformly at random whereas the size of $Y$ was varied between 0% and 25%. The MCS’s were composed by four classifiers, namely MLP, RIP, SMO, and BN. These were chosen after an experimental analysis. The evaluation results are summarized in Table 5.8. As it was expected, the classification accuracy diminishes as the level of noise increases. Interestingly, some algorithms are more noise tolerant than others. While LOG, IBK, ALR, and NB are significantly affected by only introducing 5% of noise (accuracy drops down by up to 20%), MLP and BN are able to handle noise more effectively. But note that in all the cases, the FP strategy achieves higher accuracy than all other 15 algorithms—including ALR—for all noise levels. This fact demonstrates that FP is more noise tolerant than the base classifiers and ensembles. Finally, the overall accuracy of ALR and FP is plotted in Figure 5.2.

5.3 Concluding Remarks

In this chapter, two probabilistic strategies were proposed for decision selection and fusion in multiple classifier systems. The evaluation results support the hypothesis that the proposed methods significantly improve the classification accuracy with respect to the base classifiers and other selection and fusion methods. Furthermore, an algorithm to select the
Table 5.8: Accuracy of base classifiers and ensembles with different noise levels.

<table>
<thead>
<tr>
<th>Method</th>
<th>0%</th>
<th>5%</th>
<th>10%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>90.59</td>
<td>85.4</td>
<td>77.98</td>
<td>70.14</td>
<td>65.49</td>
<td>55.11</td>
</tr>
<tr>
<td>C4.5</td>
<td>89.56</td>
<td>81.18</td>
<td>73.77</td>
<td>65.64</td>
<td>60.29</td>
<td>50.88</td>
</tr>
<tr>
<td>IBK</td>
<td>92.37</td>
<td>84.07</td>
<td>75.17</td>
<td>65.26</td>
<td>61.05</td>
<td>50.89</td>
</tr>
<tr>
<td>LOG</td>
<td>82.58</td>
<td>71.18</td>
<td>60.66</td>
<td>51.82</td>
<td>46.31</td>
<td>43.77</td>
</tr>
<tr>
<td>RIP</td>
<td>83.1</td>
<td>76.75</td>
<td>73.19</td>
<td>67.55</td>
<td>62.52</td>
<td>56.51</td>
</tr>
<tr>
<td>BN</td>
<td>88.96</td>
<td>84.16</td>
<td>77.92</td>
<td>75.48</td>
<td>69.33</td>
<td>62.07</td>
</tr>
<tr>
<td>SMO</td>
<td>86.37</td>
<td>80.6</td>
<td>75.84</td>
<td>72.14</td>
<td>64.96</td>
<td>59.03</td>
</tr>
<tr>
<td>NB</td>
<td>86.58</td>
<td>73.55</td>
<td>60.09</td>
<td>58.66</td>
<td>53.19</td>
<td>51.77</td>
</tr>
<tr>
<td>ALR</td>
<td>92.89</td>
<td>86.3</td>
<td>77.32</td>
<td>73.25</td>
<td>65.86</td>
<td>57.39</td>
</tr>
<tr>
<td>BJ48</td>
<td>88.43</td>
<td>84.52</td>
<td>78.88</td>
<td>74.74</td>
<td>68.66</td>
<td>61.77</td>
</tr>
<tr>
<td>LCA</td>
<td>90.66</td>
<td>87.99</td>
<td>79.9</td>
<td>71.85</td>
<td>66.51</td>
<td>58.96</td>
</tr>
<tr>
<td>NBC</td>
<td>78.43</td>
<td>77.21</td>
<td>75.25</td>
<td>71.77</td>
<td>66.21</td>
<td>58.21</td>
</tr>
<tr>
<td>PLU</td>
<td>90.81</td>
<td>85.63</td>
<td>79.99</td>
<td>75.93</td>
<td>69.33</td>
<td>62.73</td>
</tr>
<tr>
<td>PRD</td>
<td>90.96</td>
<td>87.03</td>
<td>80.13</td>
<td>72.75</td>
<td>69.4</td>
<td>60.88</td>
</tr>
<tr>
<td>FP</td>
<td>91.25</td>
<td>88.3</td>
<td>81.76</td>
<td>76.74</td>
<td>70.81</td>
<td>65.56</td>
</tr>
</tbody>
</table>

The best subset of base classifiers (LOC) is presented, not only reducing the complexity, but also improving the classification accuracy. More specifically, in a HAR context, the proposed strategies were demonstrated to be more effective to handle label noise.
Figure 5.2: Overall accuracy of ALR and FP with different noise levels.
Chapter 6: Conclusions and Future Work

This chapter comprises two parts: First, it summarizes the most relevant findings and results of this dissertation. Then, a number of ideas are proposed for future research consideration, in order to extend the field of Human Activity Recognition towards more realistic and pervasive scenarios.

6.1 Summary of Findings and Results

Chapter 2 presented a new two-level taxonomy of HAR systems according to their response time and learning approach. Twenty eight systems are qualitatively compared under a number of design issues such as obtrusiveness, flexibility, recognition accuracy, and energy consumption, among others. As a result, the first survey paper on Human Activity Recognition using Wearable Sensor was compiled and published [32].

In Chapter 3, Centinela [33] was presented as an effective human activity recognition system combining acceleration along with physiological signals. Five activities are recognized by Centinela: running, walking, still, walking upstairs and walking downstairs. An experimental evaluation supports the hypothesis that vital signs are beneficial to improve the recognition accuracy of human activities. The evaluation also indicates that the Additive Logistic Regression classifier using 5s time windows with 50% overlap yields the highest accuracy (i.e., up to 95.7%). To achieve the peak accuracy, time- and frequency domain features should be extracted from acceleration signals whereas two features should be calculated from physiological signals: (1) the signed magnitude of change \( \eta \) and (2) the parameter \( b \) of the line \( y = m(t) + b \) that best fits the points in the signal. Another important point of discussion is the placement of the sensor. Placing the accelerometer on the individual’s
chest had not been quite explored but it certainly avoids confusions that may arise if it is placed on the wrist [21].

In Chapter 4, the recognition of activities was taken one step further, introducing Vigilante [35, 34] as a new platform for real-time HAR. The system partially integrates the Weka API in the Android platform to enable the evaluation of a number of classification algorithms. A new library, MECLA, was also proposed to enable tree-based classification algorithms in mobile devices —which are not fully functional with WEKA. The evaluation shows that Vigilante can be effectively deployed on current cellular phones in three main regards:

- **Accuracy**, as human activities are recognized with an overall accuracy of up to 96.8%.
- **Response time**, as the total computational time required for preprocessing, feature extraction, and classification accounts for less than 8% of the window length.
- **Energy consumption**, as the application is able to run for up to 12.5 continuous hours and it enables energy savings of up to 27% with respect to a system that sends all the raw data to the server for remote processing.

Finally, Chapter 5 proposed and evaluated two new probabilistic strategies for decision selection and fusion in multiple classifier systems. The evaluation results support the hypothesis that the proposed methods significantly improve the classification accuracy with respect to the base classifiers and other selection and fusion methods. An algorithm to select the best subset of base classifiers is also included, allowing to reduce the complexity and improve the classification accuracy.

### 6.2 Future Research Considerations

In order to realize the full potential in HAR systems, some topics need further investigation. Next, a list of those topics is included.
6.2.1 Activity Recognition Datasets

The quantitative comparison of HAR approaches has been hindered by the fact that each system works with a different dataset. While in research areas such as data mining, there exist standard datasets to validate the effectiveness of a new method, this is not the case in activity recognition. Each research group collects data from different individuals, uses a different activity set, and utilizes a different evaluation methodology. In that direction, we have included various datasets publicly open to the research community which can be used as benchmarks to evaluate new approaches. Several universities and institutions have published their datasets in [104, 105, 106, 107]. Another dataset is provided by the 2011 Activity Recognition Challenge [108], in which researchers worldwide were invited to participate.

6.2.2 Composite Activities

The activities explored in this dissertation are rather simple. In fact, many of them could be part of more complex routines or behaviors. Imagine, for example, the problem of automatically recognizing when a user is playing tennis. Such activity is composed by several instances of walking, running, and sitting, among others, with certain logical sequence and duration. The recognition of these composite activities from a set of atomic activities would surely enrich context awareness but, at the same time, brings additional uncertainty. Blanke et al. [109] provide an overview on this topic and propose a solution through several layers of inference.

6.2.3 Concurrent and Overlapping Activities

The assumption that an individual only performs one activity at a time is true for basic ambulation activities (e.g., walking, running, lying, etc.). In general, human activities are rather overlapping and concurrent. A person could be walking while brushing their teeth, or watching TV while having lunch. Since only few works have been reported in this area,
we foresee great research opportunities in this field. The interested reader might refer to
the article of Helaoui et al. [110] for further information.

6.2.4 Multiattribute Classification

The purpose of a HAR system is, of course, providing feedback on the user’s activity. But, context awareness may be enriched by also recognizing user’s personal attributes. A case study could be a system that not only recognizes an individual running, but also identifies them as a female between 30 and 40 years old. We hypothesize that vital signs may have an important role in the determination of these attributes. To the best of our knowledge, there is no previous work on this topic.

6.2.5 Cost-sensitive Classification

Imagine an activity recognition system monitoring a patient with heart disease who cannot make significant physical effort. The system should never predict that the individual is sitting when they are actually running. But confusions between activities such as waking and sitting might be tolerable in this scenario. Cost-sensitive classification works exactly in that direction, maintaining a cost matrix $C$ where the value $C_{ij}$ is the cost of predicting activity $i$ given that the actual activity is $j$. The values in this matrix depend on the specific application. In prediction time, the classifier can be easily adapted to output the activity class with the smallest misclassification cost. Also, in training time, the proportion of instances can be increased for the most expensive classes, forcing the learning algorithm to classify them more accurately. Additional information on cost-sensitive classification is available in [53, 111, 4].

6.2.6 Crowd-HAR

The recognition of human activities has been somehow individualized, i.e., the majority of the systems predict activities in a single user. Although information from social networks has been shown effective to recognize human behaviors [112], recognizing collective activity
patterns can be taken one step further. If we could gather activity patterns from a significant sample of people in certain area (e.g., a city, a state, or a country), that information could be used to estimate levels of sedentarism, exercise habits, and even health conditions in a target population. Furthermore, this sort of participatory-human-centric application would not require an economic incentive method. The users would be willing to participate in the system as long as they receive information on their health conditions and exercise performance, for example. Such data from thousands or millions of users may also be used to feed classification algorithms thereby enhancing their overall accuracy.

6.2.7 Predicting Future Activities

Previous works have not only estimated activities but also behavior routines [68]. Based on this information, the system could predict what the user is about to do. This becomes especially useful for certain applications such as those based on advertisements. For instance, if the user is going to have lunch, he or she may receive advertisement on restaurants nearby.

6.2.8 User Flexibility

People certainly perform activities in a different manner due to particular physical characteristics. Thus, acceleration signals measured from a child versus an elderly person are expected to be quite different. As seen in Section 2.1.8, a human activity recognition model might be either monolithic or user-specific, each one having its own benefits and drawbacks. A middle ground to make the most of both worlds might be creating group-specific classifiers, clustering individuals with similar characteristics such as age, weight, gender, health conditions, among others. Then, our hypothesis is that a HAR system would be more effective having a recognition model for overweight young men, one for normal male children, another one for female elderly, and so forth.
References


[105] “Datasets for Human Activity Recognition from MIT Media Lab,” http://architecture.mit.edu/house_n/data/PlaceLab/PlaceLab.htm.


Appendices
Appendix A: Permissions

Thank You For Your Order!

Dear Mr. Oscar Lara Yejas,

Thank you for placing your order through Copyright Clearance Center's RightsLink service. Elsevier has partnered with RightsLink to license its content. This notice is a confirmation that your order was successful.

Your order details and publisher terms and conditions are available by clicking the link below:
http://s100.copyright.com/CustomerAdmin/PLE.jsp?IID=2012061_1340635286447

Order Details
Licensee: Oscar Lara Yejas
License Date: Jun 25, 2012
License Number: 2936961334447
Publication: Pervasive and Mobile Computing
Title: Centinela: A human activity recognition system based on acceleration and vital sign data
Type Of Use: reuse in a thesis/dissertation
Total: 0.00 USD

To access your account, please visit https://myaccount.copyright.com.

Please note: Online payments are charged immediately after order confirmation; invoices are issued daily and are payable immediately upon receipt.

To ensure we are continuously improving our services, please take a moment to complete our customer satisfaction survey.

B.1:v4.2

+1.877.622.5543 / Tel: +1.978.646.2777
customercaro@copyright.com
http://www.copyright.com
Appendix A (Continued)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license; however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © [year of original publication] IEEE.

2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].

2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.

3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
Centinela: A human activity recognition system based on acceleration and vital sign data

Óscar D. Lara A, Alfredo J. Pérez A, Miguel A. Labrador A,*, José D. Posada B

A Department of Computer Science and Engineering, University of South Florida, Tampa, Fl. 33620, United States
B Departamento de Ingeniería Electrónica, Universidad Autónoma del Caribe, Barranquilla, Colombia

ABSTRACT

This paper presents Centinela, a system that combines acceleration data with vital signs to achieve highly accurate activity recognition. Centinela recognizes five activities: walking, running, sitting, ascending, and descending. The system includes a portable and unobtrusive real-time data collection platform, which only requires a single sensing device and a mobile phone. To extract features, both statistical and structural detectors are applied, and two new features are proposed to discriminate among activities during periods of vital sign stabilization. After evaluating eight different classifiers and three different time window sizes, our results show that Centinela achieves up to 95.7% overall accuracy, which is higher than current approaches under similar conditions. Our results also indicate that vital signs are useful to discriminate between certain activities. Indeed, Centinela achieves 100% accuracy for activities such as running and sitting, and slightly improves the classification accuracy for ascending compared to the cases that utilize acceleration data only.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In the past decade, there has been a significant advance of mobile devices and sensors in regards to size, cost, and power. This has enabled new sources of data to study people’s daily activities and behaviors. Hence human-centric sensing came into picture as a promising research area in computer science [1]. Particularly, the recognition of human physical activities has become a task of high interest within the field, especially for medical, military, and security applications. For instance, patients with dementia and other mental pathologies could be monitored to detect abnormal activities and thereby prevent undesirable consequences [2]. An interactive game might also require information about which activity the user is performing in order to respond accordingly. In tactical scenarios, the soldiers’ activities along with their location may be useful to send alerts in case of danger.

All these applications need to solve the activity recognition problem, which from a practical point of view, can be defined as follows: given a time window W, defined within time instants t_s and t_e which contains a set of time series S = {S_1, . . . , S_k}, from each of the k measured attributes, the goal is to determine the activity performed during W from the predefined set of mutually exclusive activities (e.g., sitting, walking, eating, etc.). Now, recognizing human activities is not a trivial task. As a matter of fact, several challenges lie in this process, such as the selection of the attributes to be measured, the extraction of meaningful features, and the recognition of ambiguous activities. Energy consumption is also a critical issue in terms of deciding which sensors to turn on and off at any time, or setting the optimal sampling resolution [3].
Appendix B (Continued)

Most of the previously proposed schemes in activity recognition collect data from either triaxial accelerometers, video sequences [4], or environmental variables. However, little work has been reported considering vital sign data. We believe there is a noticeable relationship between the behavior of the vital signs and the physical activity. When an individual begins running, for instance, it is expected that their heart rate and breath amplitude increase. Consequently, we hypothesize that higher human activity recognition accuracy can be achieved using both acceleration and vital sign data. To illustrate this, consider the situation in Fig. 1. Data from triaxial acceleration and vital signs were recorded while a subject was ascending after walking. Note that the acceleration signals within most time intervals are very similar for both activities. Instead, the heart rate time series exhibits a very clear pattern, as a person requires more physical effort to climb stairs than to walk. This might allow us to classify said activities more accurately.

This paper presents Centinela, a human physical activity recognition system for five different activities: sitting, walking, running, ascending, and descending. The proposed methodology encompasses (1) collecting vital sign and acceleration data from human subjects; (2) extracting features from the measured attributes; (3) building supervised machine learning models for activity classification; and (4) evaluating the accuracy of the models under different parameter configurations.

The main contributions of this work are listed below:

- Centinela combines acceleration data with vital signs to achieve highly accurate activity recognition. In fact, it provides higher accuracy than other approaches under the same conditions.
- Since vital signs are not expected to change abruptly, Centinela applies structure detectors [5], i.e., linear and non-linear functions, to extract features.
- Two new features are proposed for vital signs: magnitude of change and trend, intended to discriminate among activities during periods of vital sign stabilization.
- Centinela features a portable and unobtrusive real-time data collection platform, which allows not only for activity recognition but also for monitoring health conditions of target individuals.
- Several classifiers are analyzed in the study, allowing other researchers and application developers to use the most appropriate classifiers for specific activities.

The rest of the paper is organized as follows: Section 2 analyzes the state of the art in human activity recognition. Later, Section 3 introduces the global structure of Centinela. Section 3.1 describes the data acquisition architecture, as well as the data collection protocol. Section 3.2 covers the methods applied for feature extraction, i.e., statistical, structural, and transient features. Then, Section 4 presents the methodology of the experiments and the main results. Finally, Section 5 summarizes the most important conclusions and findings.

2. Related work

Although the first works in human activity recognition (HAR) date back to the late ’90s [6], there are still many issues that motivate the development of new techniques to improve accuracy under more realistic conditions. These issues concern to four different phases: data collection, feature extraction, classification, and evaluation.

2.1. Data collection

With respect to data collection, it is crucial to make an appropriate selection of the attributes to be measured and the sensors to be used. Many previously proposed schemes use triaxial accelerometers on different parts of the body (e.g., wrist,
thigh, leg, pocket, etc.) to recognize ambulation activities (e.g., walking, running, lying, etc.) [7–12]. Other methods are based upon environmental variables and utilize microphones, light sensors, and humidity sensors, among others [13,14]. Nevertheless, little work has been done using vital sign data. Tapia et al. [15] proposed an activity recognition system that combines data from five triaxial accelerometers and a heart rate monitor. However, they concluded that the heart rate is not useful to discriminate between activities. Their argument, which is valid, is that after performing physically demanding activities (e.g., running) the heart rate remains high for a while, even if the individual is lying or sitting. To deal with this issue, Cenit handled utilizes new feature extraction techniques that allow for activity recognition during periods of vital sign stabilization.

It is also important to build an effective data collection system (i.e., hardware and software) in terms of portability, reliability, energy consumption, comfort, and cost. Some methods require four or five accelerometers in different parts of the body [12,15,16], or need the user to carry a heavy rucksack with a computer and other recording devices [14]. This might be invasive, uncomfortable, expensive, and hence not suitable for online activity recognition. Cenit requires one single sensing device, which is comfortable and unobtrusive (see Section 3.1), and a Java-enabled cellphone with Bluetooth connectivity.

2.2. Feature extraction

Existing HAR systems based on accelerometer data employ statistical feature extraction. Most of them apply either time-domain features such as mean, variance, energy, correlation between axes, etc. [11–17], or frequency-domain features, such as entropy and the coefficients of the Fourier transform. Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA) have also been applied with promising results [10], as well as autoregressive model coefficients [7]. All these techniques are conceived to handle the high variability of acceleration signals. In contrast, vital signs fluctuate smoothly and are not expected to suddenly change in short periods of time. Therefore, structure detectors [5] are utilized in this work to approximate vital sign time series by means of linear and non-linear functions. Moreover, two new features are proposed: the magnitude of change and trend of vital signs, intended to discriminate among activities during periods of vital sign stabilization.

2.3. Classification

Many classification algorithms have been applied for activity recognition: decision trees, such as C4.5 and ID3 [9,12–14,16,18]; Bayesian methods, such as Naive Bayes (NB) and Bayesian Networks (BN) [12,15,18]; Nearest Neighbor [13,18]; Fuzzy Logic [11,17]; Neural Networks [19]; and Support Vector Machines [7,8,10], among others. In this work we not only evaluate traditional classifiers, such as Naive Bayes, Bayesian Networks, C4.5, and Multilayer Perceptron, but also classifier ensembles with methods such as Bagging and Boosting. The main idea behind these techniques is to make decisions based upon the output of a set of classifiers rather than considering one single learning method. Section 4 shows the methodology and results of the classifier evaluation.

2.4. Evaluation

Two types of analyses have been proposed to evaluate activity recognition systems: subject-dependent and subject-independent evaluations [15]. In the first one, a classifier is trained and tested for each individual with his/her own data and the average accuracy for all subjects is computed. In the second one, only one classifier is built splitting the data of all individuals into a training set and a testing set.

It is important to emphasize that each person may perform activities in a different manner, which makes subject-independent analysis more challenging. In practice, a real-time activity recognition system should be able to fit any individual. It would not be convenient to train the system for each new user, especially when (1) there are too many activities; (2) some activities are not desirable for the subject to carry out (e.g., falling downstairs); or (3) the subject would not cooperate with the data collection process (e.g., patients with dementia and other mental pathologies). Thus, a subject-independent analysis, as the one presented in this paper, is preferred.

A comparison of the classification accuracy given by Cenit and other state-of-the-art approaches is included in Section 4.2.5.

3. Description of the system

Fig. 2 illustrates the process for activity recognition. First, data are collected from accelerometer and vital sign sensors, as described in Section 3.1. Then, time-domain and frequency-domain statistical feature extraction is applied to the acceleration signals (Section 3.2.1), as well as structural and transient features are extracted from vital signs (Sections 3.2.2 and 3.2.3). Next, the dataset with the extracted features is given as input to various classification algorithms and the classification accuracy of each one is calculated by means of cross validation (Section 4). Finally, the best classifier is selected as the result of a non-parametric statistical test.
Appendix B (Continued)

3.1. Data collection

Fig. 3 shows the system architecture for the data collection phase. The sensing device (see Section 3.1.1 for more details) communicates via Bluetooth with an Internet-enabled cellphone. There is a mobile application which decodes the packets and sends labeled data to the application server via the Internet. The server then receives these data and stores them into a relational database.

3.1.1. Sensing device

We are using the BioHarness™ BT chest sensor strap [20] manufactured by Zephyr shown in Fig. 3. This device features a triaxial accelerometer and allows for measuring vital signs as well. The strap is unobtrusive, lightweight, and can be easily worn by any person. The measured attributes are: heart rate, respiration rate, breath amplitude, skin temperature, posture (i.e., inclination of the sensor), electrocardiogram amplitude, and 3D acceleration. The accelerometer records measurements at 50 Hz, each one between −3g to 3g, where g stands for the acceleration due to gravity. Acceleration samples are aggregated in packets sent every 400 ms, so every packet contains twenty acceleration measurements in all three dimensions. On the other hand, the vital signs are sampled at 1 Hz, since they are not expected to change considerably in short periods of time.

In the literature, accelerometers are commonly placed either on the wrist [12,14,16], ankle [12,16], or in the trouser’s pocket [7,8,10], yet a person might be moving his/her arms or legs while being seated. This fact may introduce noise to the data, thereby causing misclassification. We believe that placing the accelerometer on the chest makes our system more noise tolerant, and our results support this hypothesis.

Please cite this article in press as: O.D. Lara, et al., Centinela: A human activity recognition system based on acceleration and vital sign data, Pervasive and Mobile Computing (2011), doi:10.1016/j.pmcj.2011.06.004
3.1.2. Mobile application

A mobile software application was built to collect training data under the Java ME platform. This allows Centinela to run on any mobile phone that supports Java, thereby avoiding the inconvenience of requiring the user to carry additional devices. The mobile application receives and decodes the raw data sent from the sensor via Bluetooth, visualizes the measurements (see Fig. 4(a)), and labels each measurement according to the option selected by the user, either: running, walking, sitting, ascending, or descending (see Fig. 4(b)). The samples are sent in real time, via UDP, to the application server, which stores the labeled data into a relational PostgreSQL database.

3.1.3. Data collection protocol

The data were collected in a naturalistic fashion, thus, no specific instructions were given to the participants. The speed, intensity, gait, and other environmental conditions were arbitrarily chosen by the subjects. Eight individuals, 7 males and 1 female, participated in this study. Their physical characteristics, namely age, weight, height, and body mass index are shown in Table 1.

Unlike accelerometer signals, vital signs do not abruptly vary after the person changes activities. On the contrary, the values of vital signs during time interval \( t \) depend of the activity during \( t_{i-1} \). Therefore, the data should be collected so that the recognition of each single activity can be independent of the previous state. If we required, for instance, the individuals to be at rest before recording each session, the system would not be trained to recognize interleaving activities! Consequently, we have collected data from subjects while performing successive pairs of activities, e.g., running before sitting, walking before descending, and so on. This was carried out for all twenty possible combinations of pairs of consecutive activities.

3.2. Feature extraction

In general, two approaches have been proposed to extract features in time series data: statistical detectors and structure detectors [5]. Statistical detectors, such as the Fourier transform and the Wavelet transform, use quantitative characteristics of the data to extract features. On the other hand, structure detectors take into account the interrelationship among data. Hence, they have been widely used for image processing and time series analysis. Due to both acceleration and physiological signals being distinct in nature, we have employed methods from statistical and structural feature extraction.

Now, to overcome the problem of detecting transitions between activities, all measured signals were divided into fixed size 50% overlap time windows [11,12]. Three different window sizes were tried: 5s, 12s, and 20s. For every time window, 90 features were extracted as follows: eight statistical features for each of the acceleration signals (i.e., 24 features), nine structural features for each of the physiological signals (i.e., 54 features), and two transient features for each of the physiological signals (i.e., 12 features). Table 2 summarizes the feature set computed from raw signals in this work. The definitions of these features are presented in the following subsections.

---

Please cite this article in press as: O.D. Lara, et al., Centinela: A human activity recognition system based on acceleration and vital sign data, Pervasive and Mobile Computing (2011), doi: 10.1016/j.pmcj.2011.06.004
Appendix B (Continued)

### Table 2

<table>
<thead>
<tr>
<th>Measured signals</th>
<th>Extracted features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Statistical</td>
</tr>
<tr>
<td>AcxX (g)</td>
<td>X</td>
</tr>
<tr>
<td>AcxY (g)</td>
<td>X</td>
</tr>
<tr>
<td>AcxZ (g)</td>
<td>X</td>
</tr>
<tr>
<td>Heart rate</td>
<td>X</td>
</tr>
<tr>
<td>Respiration rate</td>
<td>X</td>
</tr>
<tr>
<td>Breath amplitude</td>
<td>X</td>
</tr>
<tr>
<td>Skin temperature</td>
<td>X</td>
</tr>
<tr>
<td>Posture</td>
<td>X</td>
</tr>
<tr>
<td>ECG amplitude</td>
<td>X</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Function</th>
<th>Equation</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$F(t) = mt + b$</td>
<td>$[m, b]$</td>
</tr>
<tr>
<td>Polynomial</td>
<td>$F(t) = a_0 + a_1 t + \cdots + a_n t^{n-1}$</td>
<td>$[a_0, \ldots, a_n]$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$F(t) = a_0 e^{(b t + c)}$</td>
<td>$[a_0, b, c]$</td>
</tr>
<tr>
<td>Sinusoidal</td>
<td>$F(t) = a \sin(b t + c)$</td>
<td>$[a, b, c]$</td>
</tr>
</tbody>
</table>

### 3.2.1. Statistical features

Time-domain and frequency-domain features [11–17] have been extensively used to filter the relevant information of acceleration signals. In this work, eight features were calculated for all three acceleration signals (24 total features). These are: mean, variance, standard deviation, correlation between axes, interquartile range, mean absolute deviation, and root mean square, from the time domain; and, energy from the frequency domain. The interested reader might refer to [11] for the definition of all these features.

### 3.2.2. Structural features

Since vital signs have much lower variability than acceleration signals, structure detectors turn out to be a suitable approach to extract features from vital sign time series. Structure detectors use an arbitrary function $f$ with a set of free parameters $[a_0, \ldots, a_n]$ to fit the points of a given time series $S$ [5]: these parameters are, in fact, the extracted features. In order to evaluate the goodness of fit of $f$ to $S$, the sum of squared error (SSE) is computed. For each measured attribute, the goal was to find the function $f^*$ with the smallest SSE, so that the feature extraction process relies on the calculation of the free parameters of $f^*$. Table 3 summarizes the different types of functions that have been evaluated in this work.

The median of the SSE was calculated for all time windows from all six physiological signals and all four structure detectors. The median was preferred over the mean to prevent noisy instances to bias the goodness of fit of the feature detectors. From the evaluation, polynomial functions of third degree had the lowest SSE for all six vital signs. Polynomials of degree higher than three were not considered to avoid overfitting due to Runge’s phenomenon [22]. A total of nine structural features were extracted from each vital sign time window, i.e., the coefficients of the polynomials of degree one, two, and three that best fit the points in the time window.

### 3.2.3. Transient features

Consider, for instance, that someone is running for one minute and then sits for two minutes. Even though the individual is seated, the vital signs (e.g., heart rate, respiration rate, etc.) remain as if he/she was running for an interval of time that we have called the transient period. To overcome this issue, two new features are proposed in this work, the trend $\tau$, and the magnitude of change $\psi$, intended to describe the behavior of the vital signs during transient periods. The trend indicates whether the time series is increasing, decreasing, or constant. Notice that, due to the nature of the human activities considered in this work, it is expected that vital signs are either strictly increasing, strictly decreasing, or remain constant while an individual is performing one single activity.

**Definition 1. Trend.**

Let $m$ be the slope of the line that best fits the series $S$. Then, the trend $\tau(S, r)$ of $S$ is defined as follows:

$$
\tau(S, r) = \begin{cases} 
1 & \text{(increasing)} \\
-1 & \text{(decreasing)} \\
0 & \text{(constant)} 
\end{cases} \quad \text{if } |m| < |r|
$$

### Please cite this article in press as:

where $r$ is a positive real number that stands for the slope threshold. This value was set to $\tan(15^\circ)$ after doing an experimental analysis over the entire dataset. The trend can be computed in $O(1)$, given that the slope of the line that best fits the data points was calculated beforehand as one of the structural features.

Now, it is important not only to detect whether the vital signs increased or decreased, but also to measure how much they varied. For this purpose the magnitude of change feature is presented as follows:

**Definition 2.** Magnitude of change.

Let $S$ be a given time series defined from $t_{\min}$ to $t_{\max}$. Let $S_p^-$ be a subset of $S$ which contains all measurements between $t_{\min}$ and $(t_{\max} - t_{\min})p$, where $0 < p < 1$ is a percentage of the series. Let $S_p^+$ be a subset of $S$ which contains all samples between $t_{\min} + (t_{\max} - t_{\min})(1-p)$ and $t_{\max}$. Then, the magnitude of change $\kappa(S, p)$ is defined as

$$\kappa(S, p) = \max \left( \left| \max (S_p^-) - \min (S_p^-) \right|, \left| \max (S_p^+) - \min (S_p^+) \right| \right).$$

(2)

The value of $p$ was set to 0.2 after doing an experimental analysis over the entire dataset. This implies that $S^-$ is the first 20% of the series and $S^+$ would be the last 20% of the series. The purpose of the magnitude of change is to estimate the maximum deviation between the beginning and the end of the series, and it can be calculated in linear time. Fig. 5 illustrates the process of calculating this feature.

Using transient features together, our hypothesis is that a classifier could generate rules such as: if $\kappa(S_{thr}, p)$ is large and $\tau(S_{thr}, r)$ is increasing, then activity is running, for a given heart rate time series $S_{thr}$.

Even though both magnitude of change and trend are strongly related to the slope of the line that best fits a time series, they are different measures of the data shape. Section 4.2.4 analyzes the effectiveness of the two new features proposed in this work.

4. Evaluation

This section describes the methodology to evaluate the system and provides further analysis and discussion of our results and main findings.

4.1. Design of the experiments

Activity recognition was fulfilled by assessing three different datasets: the first one, $D_{acc}$, solely contains the features extracted from acceleration data; the second, $D_{sp}$, has statistical and structural features; and, the last one, $D_{ts}$, includes all features (i.e., statistical, structural and transient). This is with the purpose of measuring the impact of vital signs features in the classification accuracy. Eight classification algorithms were also evaluated for each dataset:

1. Naive Bayes (NB) [23].
2. Bayesian Network (BN) using K2 search algorithm [24].
3. J48 decision tree, which is an implementation of the C4.5 algorithm [25].
4. Multilayer Perceptron (MLP), which relies on a Backpropagation Neural Network [25].
5. Additive Logistic Regression (ALR) [26], performing Boosting with an ensemble of ten Decision Stump classifiers.
6. Bagging using an ensemble of ten Naive Bayes classifiers (BNB) and each bag having the same size than the training set.
7. Bagging using an ensemble of ten Bayesian Network classifiers (BNB) and each bag having the same size than the training set.
8. Bagging using an ensemble of ten J48 classifiers (BJ48) and each bag having the same size than the training set.
Appendix B (Continued)

We do not elaborate on the classification algorithms since they are not part of the contributions of this work. The interested reader may refer to [23–26] for a complete description of them.

The evaluation encompasses two parts: first, the selection of the best classifier(s), and later the calculation of their accuracy. This process was completed for all three datasets.

In order to determine whether a classifier is better than another, a $5 \times 2$ fold cross validation [25] was performed. In general, a $5 \times 2$ fold cross validation is preferred over other approaches (e.g., 10 fold cross validation or percentage split) because it has a smaller probability of concluding that one classifier is better than another when it is not the case. As we do not have information regarding the probability distribution of the accuracy of each classifier, the non-parametric Sign Test [27] was utilized. This test allows us to determine whether there is any statistical difference between the probability distribution functions of two independent random variables. The test is defined as follows:

**Definition 3.** Sign test.

- Let $X$, $Y$ be two independent random variables.
- Let $H_0 : P(X > Y) = P(X < Y)$ be the null hypothesis.
- Let $(x_i, y_i)$ be a set of $n$ matched pairs from $X$ and $Y$ respectively.
- Let $n'$ be the number of observations such that $x_i \neq y_i$.
- Let $T$ be the number of observations such that $x_i > y_i$. $T$ follows the binomial distribution under $H_0$ with parameters $n'$ (i.e., the number of trials) and $p = 0.5$.

The $p$-values for the test are as follows:

- **Lower:** $p_{low} = P(T \leq T_{low})$. Alternative hypothesis: $H_1 : P(X > Y) < P(X < Y)$.
- **Upper:** $p_{upper} = P(T \geq T_{low})$. Alternative hypothesis: $H_1 : P(X > Y) > P(X < Y)$.
- **Two-sided:** $p = 2 \min(p_{low}, p_{upper})$. Alternative hypothesis: $H_1 : P(X > Y) \neq P(X < Y)$.

In our case, the random variables $X$ and $Y$ are the values from the accuracy of two classifiers, and there are five matched pairs for all different random seeds. A way to extend this test to deal with $n$ random variables (since we are considering eight algorithms) is to make a $k$-round binary tournament among pairs of classifiers. If the null hypothesis is rejected, the best classifier (i.e., the winner) goes to the next round, and the other one (i.e., the loser) is discarded. If the null hypothesis is not rejected, both classifiers pass to the next round. The process repeats until no more classifiers can be discarded. At the very end, the classifiers which did not lose any match are selected as the best classifiers.

All the classification algorithms were tested in the Waikato Environment for Knowledge Analysis (WEKA) [28]. This is a well-known software tool developed by the University of Waikato, New Zealand, which allows for easy evaluation of machine learning algorithms. Three window sizes were evaluated, namely 5s, 12s, and 20s. The significance level for all sign tests was fixed to $\alpha = 0.05$.

4.2. Results

An interesting fact in machine learning is that the performance of a classification algorithm depends on which dataset it is applied to. As we are to decide on whether vital signs allow for more accurate classification, it is required to determine the best classifier for each dataset.

4.2.1. Dataset with features from vital signs and acceleration

The best classifiers on $D_{vis}$, according to the $5 \times 2$ cross validation tournament, are shown in Table 4, for all three different window sizes, and five random seeds $s$: 1, 128, 255, 1023, and 4095. As a notation, the name of the classifiers will be henceforth accompanied with the window size written as superscript and the dataset as subscript. For example, $\text{ALR}_{vis}^{12}$ stands for ALR over the $D_{vis}$ dataset using 12s time windows.

In order to select the most appropriate window size, the same concept of $k$-rounds binary tournament of sign tests was applied among the best eight classifiers. As a result, four of them were selected: $\text{ALR}_{vis}^{12}$, $\text{BBN}_{vis}^{12}$, $\text{B46}_{vis}^{12}$, and $\text{ALR}_{vis}^{12}$. Notice that having time windows as long as 20s considerably affected the accuracy of all the classifiers. This was expected since, in such a case, more than one activity might be performed within one single time window [29].

Let us emphasize that the $5 \times 2$ fold cross validation is not intended to measure classification accuracy but to discriminate whether or not there is a statistically significant difference among them! This is because a $5 \times 2$ fold cross validation only uses half of the data as training set. Now, the actual accuracy of the four classifiers is measured by a $5 \times 10$ fold cross validation for each activity. The average accuracies are shown in Table 5. Classifiers using time windows of five seconds had the lowest overall accuracy. This is reasonable since vital signs cannot properly describe activity patterns in time intervals as short as five seconds. Instead, Table 5 suggests that features extracted from vital signs within a window of 12s allow the system to discern among activities that might be ambiguous for accelerometers, such as walking and ascending (see Fig. 1).

Although there is no sufficient statistical evidence to assure that any of the classifiers is better than the others, we have chosen 12s as the best window size, and $\text{ALR}$ as the best classifier for the $D_{vis}$ dataset. This is not only because it reaches the highest accuracy in the $5 \times 10$ fold cross validation, but also because of how each activity was classified. Observe that $\text{ALR}_{vis}^{12}$ (i.e., ALR with 12s time windows over the $D_{vis}$ dataset) classified activities such as running, ascending, and descending with the highest accuracy. For real applications in health care or tactic scenarios, it may be more important to detect these types of activities.

Please cite this article in press as: Ó.D. Lara, et al., Continus: A human activity recognition system based on acceleration and vital sign data, Pervasive and Mobile Computing (2011), doi:10.1016/j.pmcj.2011.08.004

129
Appendix B (Continued)

Table 4
Percentage classification accuracy for the best classifiers given by the 5 × 2 fold cross validation on D_{amb}. Five different random seeds s were utilized.

<table>
<thead>
<tr>
<th>Activity</th>
<th>s_1</th>
<th>s_2</th>
<th>s_3</th>
<th>s_4</th>
<th>s_5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALR^{a}_{60}</td>
<td>89.22</td>
<td>89.49</td>
<td>91.91</td>
<td>94.34</td>
<td>92.18</td>
<td>91.98</td>
</tr>
<tr>
<td>BBN^{a}_{60}</td>
<td>90.84</td>
<td>90.57</td>
<td>91.64</td>
<td>90.83</td>
<td>89.76</td>
<td>90.57</td>
</tr>
<tr>
<td>BBN^{a}_{20}</td>
<td>88.41</td>
<td>90.03</td>
<td>93.53</td>
<td>87.6</td>
<td>88.14</td>
<td>80.54</td>
</tr>
<tr>
<td>ALR^{a}_{20}</td>
<td>92.857</td>
<td>92.86</td>
<td>87.857</td>
<td>95</td>
<td>88.57</td>
<td>91.43</td>
</tr>
<tr>
<td>BBN^{a}_{10}</td>
<td>84.286</td>
<td>87.86</td>
<td>89.286</td>
<td>91.43</td>
<td>91.71</td>
<td>89.82</td>
</tr>
<tr>
<td>BBN^{a}_{40}</td>
<td>80.822</td>
<td>84.931</td>
<td>84.931</td>
<td>80.82</td>
<td>82.19</td>
<td>82.74</td>
</tr>
<tr>
<td>ALR^{a}_{40}</td>
<td>86.301</td>
<td>89.823</td>
<td>80.822</td>
<td>87.67</td>
<td>86.30</td>
<td>84.38</td>
</tr>
<tr>
<td>BBN^{a}_{80}</td>
<td>82.192</td>
<td>83.562</td>
<td>83.562</td>
<td>87.67</td>
<td>83.56</td>
<td>84.11</td>
</tr>
</tbody>
</table>

Table 5
Per-class mean percentage accuracy of the 5 × 10 fold cross validation among the best classifiers for the D_{amb} dataset.

<table>
<thead>
<tr>
<th>Activity</th>
<th>ALR^{a}_{60}</th>
<th>BBN^{a}_{60}</th>
<th>BBN^{a}_{20}</th>
<th>ALR^{a}_{20}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>92.98</td>
<td>97.62</td>
<td>94.72</td>
<td>94.1</td>
</tr>
<tr>
<td>Running</td>
<td>98.56</td>
<td>98.8</td>
<td>98.32</td>
<td>100</td>
</tr>
<tr>
<td>Ascending</td>
<td>83</td>
<td>69.24</td>
<td>85.56</td>
<td>90.84</td>
</tr>
<tr>
<td>Descending</td>
<td>91.06</td>
<td>80.4</td>
<td>89.52</td>
<td>91.36</td>
</tr>
<tr>
<td>Sitting</td>
<td>100</td>
<td>95.58</td>
<td>96.36</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>93.12</td>
<td>88.328</td>
<td>94.24</td>
<td>95.7</td>
</tr>
</tbody>
</table>

Table 6
Percentage classification accuracy for the best classifiers given by the 5 × 2 fold cross validation using accelerometer data only. Five different random seeds s were utilized.

<table>
<thead>
<tr>
<th>Activity</th>
<th>s_1</th>
<th>s_2</th>
<th>s_3</th>
<th>s_4</th>
<th>s_5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP^{a}_{60}</td>
<td>92.45</td>
<td>91.11</td>
<td>94.61</td>
<td>88.14</td>
<td>92.99</td>
<td>91.86</td>
</tr>
<tr>
<td>ALR^{a}_{60}</td>
<td>88.95</td>
<td>91.37</td>
<td>90.84</td>
<td>92.45</td>
<td>91.91</td>
<td>91.105</td>
</tr>
</tbody>
</table>

4.2.2. Dataset with features from acceleration only

The same procedure was carried out over the dataset that only contains features from the accelerometer data. That is, a tournament of 5 × 2 fold cross validations and sign tests for all eight algorithms. Table 6 contains the results from the best classifiers given by the 5 × 2 fold cross validation over the D_{amb} dataset, namely ALR^{a}_{60} and MLP^{a}_{60}. Their accuracy was computed by means of a 5 × 10 fold cross validation (see Fig. 6).

4.2.3. Analyzing the impact of vital signs

To quantify the improvement achieved by incorporating vital sign data, the best classifiers for each dataset are now compared. Fig. 6 summarizes the classification accuracy per activity for ALR^{a}_{20}, ALR^{a}_{40}, and MLP^{a}_{60}. Note that ALR^{a}_{20} reached the highest overall accuracy (i.e., 95.7%) perfectly classifying activities such as sitting and running. The activity labeled as ascending reported the most significant improvement (between 10% and 13%). This was expected as acceleration signals are similar for ascending and walking whereas vital signs provide more clear patterns to distinguish between these activities (see Fig. 1).

Notice that MLP^{a}_{60} yields higher mean accuracy than ALR^{a}_{20} for descending (roughly 6%). This brings a new point to the discussion: depending on the application and the activities that are to be recognized, it might (or might not) be useful to consider vital sign data to recognize human activities. According to our results, if the target activities are descending or walking, the data from accelerometers would be sufficient to discover activity patterns. On the other hand, if activities such as running, sitting, or ascending need to be recognized, vital signs would definitely provide the system with a more reliable output.

4.2.4. Analyzing the impact of transient features

To evaluate the effectiveness of transient features, an additional 5 × 2 fold cross validation tournament was applied to a new dataset D_{trans} (where the subscript trans stands for no transient) which only includes statistical features from acceleration data and structural features from vital signs. After evaluating all eight classification algorithms and all three window sizes, two classifiers were chosen, ALR^{a}_{60} and ALR^{a}_{40}. These are now compared to ALR^{a}_{20} (which does include transient features), and the results are in Fig. 7. Despite the overall accuracy improvement was between 3% and 4%, transient features enhanced between 4% and 10% for ascending. We believe this improvement is worthwhile as transient features are inexpensive, computationally speaking. Finally, Table 7 shows the average of the confusion matrices from the 5 × 10 fold cross validation.
using the ALR\textsuperscript{2P} classifier. Confusions are, on average, less than 5% and only among three activities: walking, ascending, and descending. This is reasonable since these three activities might have similar patterns depending on the intensity at which they are performed by the individual.

### 4.2.5. Centinela vs. other state-of-the-art approaches

It is worth mentioning that we cannot directly compare Centinela to all other HAR systems. This is mainly because each approach carries out a different experimental setup in terms of (1) the physical characteristics of the individuals; the data were collected from, (2) the activities to be recognized, and (3) the evaluation methodology. However, to have a general idea...
Appendix B (Continued)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>94.28</td>
<td>92*</td>
<td>94.43</td>
<td>90.3</td>
</tr>
<tr>
<td>Running</td>
<td>100</td>
<td>93*</td>
<td>93.89</td>
<td>87.01</td>
</tr>
<tr>
<td>Descending</td>
<td>92.84</td>
<td>68*</td>
<td>84.12</td>
<td>88.15</td>
</tr>
<tr>
<td>Ascending</td>
<td>91.36</td>
<td>67*</td>
<td>86.42</td>
<td>92.5</td>
</tr>
<tr>
<td>Sitting</td>
<td>100</td>
<td>90*</td>
<td>99.15</td>
<td>N/A</td>
</tr>
<tr>
<td>Total</td>
<td>95.7</td>
<td>92.8*</td>
<td>93.91</td>
<td>89.74</td>
</tr>
</tbody>
</table>

of the benefits provided by Centinel. Table 8\(^1\) is presented to compare our system to three others working with the same set of activities.

Firstly, eWatch is introduced in [13] as an online activity recognition system which embeds sensors and a microcontroller within a device that can be worn as a sport watch. Four sensors are included, namely accelerometer, light sensor, thermometer, and microphone. Although eWatch features up to 92.5% overall accuracy, it achieves less than 70% of accuracy for activities such as descending and ascending. Centinel reduces the misclassification of these activities by considering vital signs and it reaches 91.36% and 92.84% respectively. Also, in eWatch, data were collected under controlled conditions, thus, a lead experimenter supervised and gave specific guidelines to the subjects on how to perform the activities [13]. In 1999, Foerster et al. [6] demonstrated 95.6% of accuracy for ambulation activities in a controlled data collection experiment, but in a natural environment, the accuracy dropped to 66%.

Secondly, the system proposed in [5] uses HAAR filters to extract features and the C4.5 algorithm for classification purposes. In activities such as running, walking and sitting, their results are fairly close to Centinel's; yet for ascending and descending, Centinel is slightly better. Furthermore, only four individuals with unknown physical characteristics participated in the study presented in [9]. Collecting data from such number of people might be insufficient to provide flexible recognition of activities on new users.

Finally, the system presented in [30] applies Hidden Markov Models and Neural Networks resulting in almost the same accuracy than Centinel's for ascending and descending. But, Centinel recognizes the activities running and walking more accurately. In addition, data collected in [30] are from one single individual, which implies that a subject-dependent analysis was performed (refer to Section 2.4 for the definition and disadvantages of a subject-dependent analysis).

In the present work, all the data were collected under naturalistic conditions and a subject-independent analysis was applied for the evaluation. Next, Centinel is qualitatively compared to other approaches that recognize a different set of activities:

In 2002, Randel et al. [19] introduced a system to recognize ambulation activities which makes use of Root Mean Square for feature extraction and Backpropagation Neural Networks for classification. The authors claim to have reached up to 95% of accuracy but also emphasize that results were analyzed after further person specific training. This implies a subject-dependent analysis which, again, might not be convenient for real applications. Additionally, the paper does not include information regarding the characteristics of the subjects, the data collection protocol, and the confusion matrix.

In [15], the authors claim that the average classification accuracy for ambulation activities is 94.6%, but this is only for subject-dependent analysis. They hardly reach 56% of accuracy in the subject-independent evaluation. The same situation occurs in the system proposed in [18]. They compare different classification algorithms to recognize ambulation activities with a 95% subject-dependent accuracy, but only reach 86% of accuracy for the subject-independent analysis.

Emres et al. [16] developed an online system that reaches 94% overall average accuracy but they only applied a subject-dependent evaluation. Besides, their data were collected from only three subjects. Kao et al. [17] also present an online system with 94.71% overall accuracy, but they incorporate other activities such as hitting, knocking, working at PC, and brushing teeth. The activities descending and ascending, included in this work, are not considered there. He et al. [7,8,10] achieved up to 97% of accuracy but only considered four activities: running, being still, jumping, and walking. These activities are quite different in nature, which considerably reduces the level of uncertainty thereby enabling higher accuracy. In this work, we consider other activities such as ascending and descending stairs which open new possibilities for real applications and require a higher level of discrimination.

In 2010, Berchtold et al. introduced Actiserv as an activity recognition service for mobile phones [31]. They make use of a fuzzy inference system to classify daily activities, achieving up to 97% of accuracy. However, this requires a runtime duration in the order of days! When their algorithms are executed to meet a feasible response time, the accuracy drops to 71%. Actiserv can also reach up to 90% after personalization, in other words, a subject-dependent analysis.

The approach proposed in [11] exhibits high recognition accuracy (about 93%). Nonetheless, all the data were collected inside the laboratory, under controlled conditions.

---

\(^1\) Values marked with an asterisk (*) are approximated. They were obtained from a chart included in [13].
4.3. Beyond the recognition of human activities

Centinela can be visualized as a general tool for pattern recognition in time series data. As a matter of fact, it could be extended to provide inference not only on the individual’s activity, but also their gender and other personal information. While Centinela can accurately make statements such as the individual is running, it would be even more useful to deliver additional information, such as a female individual between 130 and 150 pounds is running. However, in order to achieve gender, weight, and activity recognition, these new attributes have to be included as goal attributes in the classification context as well. And, in this case, instead of a single attribute classification problem, we would be dealing with a multiattribute classification problem (activity, gender, and weight). As most classification algorithms only support a single class attribute, this problem becomes quite challenging. One possible solution might be to create a composite class attribute whose domain is the Cartesian product of the atomic class attributes (i.e., all possible combinations of activities, weights, and genders). Of course, many more confusions are expected due to additional uncertainty being introduced to the system. Moreover, it would be required to acquire greater amount of data from individuals that represents the entire spectrum of possible values, i.e., females and males, from all possible weights, heights, ages, and so forth. Such a multiattribute classification problem is beyond the goals of this paper, but is part of our current research work.

5. Conclusions

This paper presents Centinela, a human activity recognition system based upon acceleration and vital sign data. An extensive evaluation was performed for three feature sets (i.e., statistical, structural, and transient), eight classification algorithms, and three different window sizes. Overall, the highest mean accuracy achieved was 95.7% for the Additive Logistic Regression algorithm with a window size of 12s and considering both vital signs and acceleration data. This fact supports the hypothesis that vital signs together with acceleration data can be useful for recognizing certain human activities more accurately than by considering acceleration data only. There are some activities, however, for which acceleration data are enough to perform accurate classification. Ensembles of classifiers turned out to have the highest accuracy, yet they require more training and testing time. This introduces new challenges to achieve online activity recognition. Another important point of discussion is the placement of the sensor. We believe that placing the accelerometer on the chest of the person avoids confusions that may arise if it is placed on the wrist [17]. As a matter of fact, Centinela reaches 92.84% of accuracy with acceleration data only, which is better than most of the previously proposed approaches. Centinela also features a portable and unobtrusive real-time data collection platform, which allows not only for activity recognition but also for monitoring health conditions of target individuals.

References


Please cite this article in press as: O.D. Lara et al., Centinela: A human activity recognition system based on acceleration and vital sign data, Pervasive and Mobile Computing (2011), doi:10.1016/j.pmcj.2011.08.004

133
Appendix B (Continued)


Appendix C: A Mobile Platform for Real-time Human Activity Recognition

A Mobile Platform for Real-time Human Activity Recognition

Óscar D. Lara and Miguel A. Labrador
Department of Computer Science and Engineering
University of South Florida, Tampa, FL 33620
olarayej@mail.usf.edu, labrador@cse.usf.edu

Abstract—Context-aware applications have been the focus of extensive research yet their implementation in mobile devices usually becomes challenging due to restrictions in regards to processing power and energy. In this paper, we propose a mobile platform to provide real-time human activity recognition. Our system features (1) an efficient library, MECLA, for the mobile evaluation of classification algorithms; and (2) a mobile application for real-time human activity recognition running within a Body Area Network. The evaluation indicates that the system can be implemented in real scenarios meeting accuracy, response time, and energy consumption requirements.

Index Terms—Body Area Networks; Human-centric sensing; Machine learning; Mobile applications.

I. INTRODUCTION

Providing accurate and opportune information on people’s activities and behaviors is one of the most important and well studied tasks in pervasive computing. Imumerable applications can be visualized for medical, security, entertainment, and tactical purposes. In recent years, the prominent development of sensing devices (e.g., accelerometers, cameras, GPS, etc.) has facilitated the process of measuring attributes related to the individuals and their surroundings. In addition, most sensors are nowadays equipped with communication capabilities which allow their integration with other mobile devices within Personal Area Networks (PANs) or Body Area Networks (BANs). However, most applications require much more than simply collecting measurements from variables of interest. In fact, additional challenges for enabling context awareness involve the design of techniques to perform data analysis and inference, as the raw data (e.g., acceleration signals or electrocardiogram) provided by the sensors are, in many cases, useless.

Even though the first works in human activity recognition (HAR) date back to the late 90’s [1], there are still significant research challenges within the field. In this work, we concentrate on one of the most relevant ones: the implementation of a HAR system in a mobile phone. This task becomes difficult because of the energy and computational constraints present in the devices. Such limitations are particularly critical for activity recognition applications which entail high demands due to decoding, feature extraction, classification, and transmission of large amounts of raw data. Furthermore, to the best of our knowledge, available machine learning API’s such as WEKA [2] and JDM [3] are not supported by current mobile platforms. This fact accentuates the necessity of an efficient mobile library for evaluating machine learning algorithms and implementing HAR systems in mobile devices.

At the same time, we foresee that a mobile HAR system will bring important advantages and benefits. For instance, a HAR system running in a BAN should substantially reduce energy expenditures, as raw data would not have to be continuously sent to a server for processing. The system would also become more robust and responsive because it would not depend on unreliable wireless communication links, which may be unavailable or error prone; this is particularly important for medical or military applications that require real-time decision making. Finally, a mobile HAR system would be more scalable since the server load would be alleviated by the locally performed feature extraction and classification computations.

In this paper, we introduce a mobile framework in support of real-time human activity recognition under the Android platform—reported as best-selling smartphone platform in 2010 by Canalys [4]—to address previously mentioned issues. A mobile application was implemented on top of Centinea [5], a HAR system based on acceleration and physiological signals to automatically recognize physical activities. The application utilizes the C4.5 classification algorithm as a proof of concept. The implementation and evaluation of our framework present the following main results and contributions:

- A library for the mobile evaluation of classification algorithms (MECLA) was successfully implemented.
- An application for real-time HAR was implemented in an Android cellular phone. It uses MECLA and supports multiple sensing devices integrated in a BAN.
- The evaluation shows that the system can be effectively deployed in current cellular phones. Indeed, the application can run for up to 12.5 continuous hours with a response time of no more than 8% of the window length and an overall accuracy of 92.6%.
- Different users with diverse characteristics participated in training and testing phases, assuring flexibility to support new users without the need of re-training the system.

The rest of the paper is organized as follows: Section II analyzes the state of the art in human activity recognition and cellular phone implementations. Later, Section III describes the design of the system. Then, Section IV presents the evaluation methodology and main results. Finally, Section V summarizes the major conclusions and findings.

II. RELATED WORK

The recognition of daily activities using accelerometer data has been the focus of extensive research [6]. Nonetheless,
many systems rely on sensors placed in different parts of the body or need the user to carry computers and other devices [7]. This might be invasive, uncomfortable, expensive, and therefore, not suitable for practical activity recognition systems. Our proposed system only requires a single sensing device, which is comfortable and unobtrusive, as well as an Android cellular phone with Bluetooth connectivity. Moreover, we also consider physiological signals, in view of their usefulness to improve activity recognition accuracy [5].

Another important aspect is the data collection protocol followed by the individuals. In 1999, Foerster et al. [1] demonstrated 95.6% of accuracy for ambulation activities in a controlled data collection experiment, but in a natural environment, the accuracy dropped to 66%! In this work, we have carried out a naturalistic data collection procedure. Human activity recognition is a well studied field yet very few works have successfully been deployed in mobile phones. In 2010, Berchtold et al. introduced ActiServ as an activity recognition service for mobile phones [8]. They make use of a fuzzy inference system to classify daily activities, achieving up to 97% of accuracy. Nevertheless, it requires a runtime duration in the order of days! When their algorithms are executed to meet a feasible response time, the accuracy drops to 71%. ActiServ can also reach up to 90% after personalization, in other words, a subject-dependent analysis (i.e., the system needs to be re-trained for new users). Brezemes et al. [9] proposed a mobile application for HAR under the Nokia platform; but they used the k-nearest neighbors classifier, which is impractical for mobile phones as it needs the entire training set—which can be fairly large—to be stored in the device. Besides, their system requires each new user to collect additional training data in order to obtain accurate results. Finally, Riboni et al. [10] presented COSAR, a framework for context-aware activity recognition using statistical and ontological reasoning under the Android platform. The system recognizes a number of activities very accurately yet it only supports statistical features and the multivariate logistic regression classifier. In this work, we have included a variety of classification and feature extraction methods, providing a wider range of mobile pattern recognition applications. Further, we have studied the energy consumption and response time of the system, something not formally evaluated in any of the previously described works.

### III. DESCRIPTION OF THE SYSTEM

The mobile application has three different user profiles: trainer, tester, and administrator. A trainer is a regular user whose activities are to be monitored and recognized. A trainer is, as indicated by its name, intended to collect training data that can be used to build additional classification models. In this profile, the user is also required to enter the real performed activity as a ground truth. Finally, the administrator is able to do training, testing, and modifying the application settings. Figure 1 shows the main components of the system and their interrelationships. First, the communication and sensing modules allow for collecting raw data from the sensing devices. These data are decoded and organized in time windows by the data preprocessing component. Then, the feature extraction module extracts statistical and structural features from each time window, producing a feature set instance which is later evaluated by the classification module. The output of the classifier is indeed the recognized activity, displayed by the visualization module and sent to the server for further analysis and historical querying. When the system is working on training mode, all raw data are also sent to the server. Next, we will elaborate on the mobile application components.

#### A. Sensing devices

The system currently supports three sensing devices, namely the phone GPS, phone accelerometer, and the BioHarness BT chest sensor strap, manufactured by Zephyr. The strap is unobtrusive, lightweight, and can be easily worn by any person. It allows for measuring the person’s heart rate, respiration rate, breath amplitude, skin temperature, posture (i.e., inclination of the sensor), electrocardiogram amplitude, galvanic skin response, $SpO_2$, and 3D acceleration. The strap accelerometer records measurements at 50 Hz, within a $-3g$ to $3g$ range, where $g$ stands for the acceleration of gravity. On the other hand, physiological signals are sampled at 1 Hz since they are not expected to change much in short periods of time.

#### B. Communication

The communication module encompasses three levels: (1) receiving raw data from the sensors (via Bluetooth), (2) sending raw data or activity results to the server (via TCP/IP); and (3) querying the database (via HTTP servlets). In the first level, three different types of packets are received from the sensing device: acceleration, vital signs, and electrocardiogram. In the second level, these packets are aggregated and sent to the application server, which decodes and stores them into the
Appendix C (Continued)

database. Finally, in the third level of communication, HTTP

servlets allow for validating user credentials and querying

the list of activities to be recognized, the list of features to be

extracted, as well as the classification models to be used.

C. Sensing and data preprocessing

The sensing component manages and synchronizes the flow

of data from all sensing devices, i.e., accelerometer, GPS, and

the BioHarness strap. Figure 2 illustrates the interrelationship

among the classes in this module. Sensors are represented as

entities that extend from the abstract class Sensor and imple-

ment methods to connect, read data, and finalize. Additionally,

they periodically report measurements to the SensorManager

class through the Observer-Observerable pattern [11]. With this

model, new sensors can be easily incorporated to the system

and they are able to work concurrently, as the class Sensor

also implements the Java Runnable interface.

Every piece of raw data is represented as a packet, which

might contain one or several samples, according to the sensor

specifications. Each specific type of packet extends from the

abstract Packet class, and implements a particular decode

method. The SensorManager class receives all packets and

d-controls the data flow to achieve the recognition of activities.

D. Feature extraction

This component provides an efficient implementation of

statistical and structural feature extraction methods for nine

attributes: heart rate, respiration rate, breath amplitude, skin

temperature, posture, ECG amplitude, and 3D acceleration.

The methods are briefly introduced below yet the interested

reader might refer to [5], [6] for more details.

1) Statistical features: These are mean, variance, standard

deviation, correlation between axes, interquartile range, mean

absolute deviation, root mean square, and energy. They were

applied to acceleration signals (i.e., 24 features in total).

2) Structural features: Structure detectors use an arbitrary

function \( f \) with a set of free parameters \( \{a_0, ..., a_n\} \) to fit the

points of a given time series \( S \); these parameters are, in fact,

the extracted features. This fitting process is done by means

of the Least Squares algorithm. For each of the vital sign time

windows, the goal is to find the function \( f^* \) with the smallest

sum of squared error (SSE). In our previous study [5], we

found that polynomial functions had the lowest SSE using

degrees one, two, and three. Finally, we considered two more

features, the trend, and the magnitude of change, intended

to describe the behavior of the vital signs during intervals

of vital sign stabilization. Structural features are applied to

physiological signals (i.e., a total of 66 features).

E. Classification

Many classification algorithms have been employed in activity

recognition: C4.5 [7], Fuzzy Logic [6], and Support Vector

Machines [12], among others. Nevertheless, implementing

them in a cellular phone turns out to be a hard task. Although

there exist Java API’s to train and evaluate classification

models, such as WEKA [2] and JDM [3], these are not suitable

for mobile phones. This is mainly because some classes and

interfaces required by them, such as java.lang.Cloneable, are

not available in the Android platform. Therefore, we wrote

our own library, MICLA, to evaluate classification models

on the phone. Our current implementation supports decision

trees, thereby enabling the use of C4.5, IDS, Decision Stump,

Random Tree, and M5, among other classification algorithms.

We are currently working on integrating additional algorithms
to the library. More on MECLA can be found in [13].

The design of the classification module is shown in Figure 3.

Nodes correspond to features (e.g., mean acc. in X) whereas

edges define relations of an attribute with a numeric or nomi-
nal value (e.g., mean acceleration in X is greater than 2.382g).

The method evaluate of the class Edge returns whether or not a
given value fulfills the edge’s relational condition. This method

is utilized by class Node’s evaluate method, which returns the

next child node to be visited given a particular attribute value.

In this way, the evaluation of a feature vector (i.e., an instance

of the class Featureset) starts at the root node and follows

the corresponding branches according to the output of the node

evaluation methods. The process stops when the current node
Appendix C (Continued)

does not have any children, and the associated activity class (e.g., running, walking, etc.) is returned.

All classification models were initially trained in WEKA using the C4.5 algorithm under different parameter configurations (i.e., window sizes and feature extraction methods). The alphanumeric representations of all decision trees, as given by the WEKA output, were stored in our database. An HTTP servlet allows the mobile application to query a decision tree model in accordance to the parameter values set by the user. We used a recursive algorithm to build a DecisionTree object (i.e., nodes and edges) based upon the alphanumeric representation retrieved by the servlet.

**F. Feature selection**

Even though the classification models were trained with a total of 90 features, only a subset of them are used. That depends on the window size, the enabled sensors, and the data themselves. Hence, the mobile application only extracts the features that are actually part of the decision tree. For this purpose, another HTTP servlet retrieves the list of features to be extracted given the window size and the active sensors.

**G. Visualization**

Figure 4 displays the two main screens, namely the monitor and the configuration. The former displays the sensor data (e.g., heart rate, respiration rate, skin temperature, etc.), the current user’s activity, and the data collection time in real-time. The latter, allows the administrator to set parameters such as the window size, number of aggregated packets, maximum number of Bluetooth reconnections, time between Bluetooth reconnections, and feature extraction methods. It also permits turning on and off each individual sensor, as well as enabling or disabling data transmission to the server.

**IV. EVALUATION**

The mobile application was tested on an HTC Evo 4G mobile phone, being compatible with Android 2.1 or higher. The classification model (shown in Figure 5) was generated by the C4.5 algorithm using 5-second-long time windows with 50% overlap. In [5], the reader may find the data collection protocol and the physical characteristics of the individuals who collected training data. As a proof of concept, in the present study, three activities were considered, namely running, walking, and sitting. Two new users (a male and a female), who did not participate in the training phase, performed each activity in a sequential fashion during approximately 5 minutes. A total of 360 instances (i.e., time windows) were classified. The evaluation encompasses three main aspects: accuracy, response time, and energy consumption.

**A. Accuracy**

To avoid potential mistakes when labeling the data, the assessment of the classification accuracy was done programatically. In training mode, each user set the real activity as the ground truth, and performed said activity for certain amount of time. Meanwhile, the application computes the classified activity for each time window. With these data, the confusion matrix is calculated and displayed on the phone. The results are shown in Figure 6, where columns correspond to the real activity and rows represent the classified activity. Precision and recall were also calculated for each activity. The overall accuracy was 92.64%.

![Confusion Matrix](image)

**Fig. 6.** Confusion matrix.
Appendix C (Continued)

B. Response time

The response time was measured for each time window as the total time spent by (1) preprocessing, (2) feature extraction, and (3) classification. The reader may notice that the decision tree shown in Figure 5 is fairly small and only involves four features. In order to demonstrate the feasibility of the application under more challenging scenarios, we have also measured the response time for two larger decision trees that use different window sizes and feature sets. The average response times for all trees, after issuing 100 time windows, are shown in Table I.

<table>
<thead>
<tr>
<th>Window length</th>
<th>Features</th>
<th>Nodes</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>5s</td>
<td>4</td>
<td>9</td>
<td>96.56 ms</td>
</tr>
<tr>
<td>5s</td>
<td>10</td>
<td>47</td>
<td>293.35 ms</td>
</tr>
<tr>
<td>12s</td>
<td>5</td>
<td>22</td>
<td>490.38 ms</td>
</tr>
</tbody>
</table>

The most demanding stage was the feature extraction which, on average, consumed 60% of the total response time. This was expected as the feature extraction algorithm runs in $O(n)$, where the number of samples $n$ can be either 250 or 600 for window lengths of 5s and 12s, respectively. On the other hand, the classification stage only consumed about 12% of the time, as it runs in $O(\log n)$, where the number of nodes $n$ is between 9 and 47. Hence, the window size and the number of features have more significant impact in the overall response time than the number of nodes. Note that, in the worst case, the response time is less than 8% of the window length, confirming that our system can be deployed in current cellular phones.

C. Energy consumption

Our hypothesis is that executing activity recognition locally in the phone—rather than sending all raw data to the server—is effective to save energy. In order to prove it, three cases were considered: (1) recognizing activities locally in the phone without sending raw data to the server; (2) sending all raw data to the server to recognize the activities remotely; and (3) not running the application to measure the energy consumed by the operating system and other phone services alone. The classes Intent and BatteryManager of the Android API were used to measure the battery charge difference after three continuous hours of use. Then, energy consumption was estimated given that the phone’s battery works at 3.7 V and has a total charge of 1500 mAh. The results, shown in Table II, indicate that performing local activity recognition allows for increasing the application lifetime in 25%. More precisely, the system can run for up to 12.5 hours in case 1, versus 9.38 hours in case 2.

<table>
<thead>
<tr>
<th>Case</th>
<th>Charge diff.</th>
<th>Current</th>
<th>Power</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.34 Ah (24%)</td>
<td>0.12 A</td>
<td>0.44 W</td>
<td>495.2 J</td>
</tr>
<tr>
<td>2</td>
<td>0.06 Ah (24%)</td>
<td>0.10 A</td>
<td>0.29 W</td>
<td>396.5 J</td>
</tr>
<tr>
<td>3</td>
<td>0.03 Ah (2%)</td>
<td>0.01 A</td>
<td>0.01 W</td>
<td>309.6 J</td>
</tr>
</tbody>
</table>

Extended to use always the local recognition, the estimated energy consumption after executing the application for three hours is: 495.2 J for case 1, 396.5 J for case 2, and 309.6 J for case 3. This result becomes remarkable as case 2 was already using data aggregation as an energy saving mechanism.

VI. ACKNOWLEDGEMENTS

This work was partially funded by the National Science Foundation under grant No. 1062160. Special thanks to Elizabeth Rodriguez, Aaron Young, and Jose Cadena, who worked on the development of the mobile application, and Anamaria Bejarano for reviewing the manuscript.

REFERENCES