Activity Identification for Gait Recognition Using Mobile Devices

Gazmend Bajrami

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2011

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Activity Identification for Gait Recognition Using Mobile Devices

Gazmend Bajrami

June 2011

Abstract

The need to increase the security measures and make the users of different devices feel safer has led the researchers focus on finding new security mechanisms with better performance. A field where not much has been done so far is finding a solution is keeping the data secured by ensuring that only the authorized user can access the data. Even though gait recognition is an important element in this project and in the entire authentication process the issue is that it is not the only element in a successful implementation of the project. This is due to different routine activities people do daily. A solution to this issue could be activity recognition that would reduce the disadvantages of gait recognition by identifying the activities of a person continuously. Activity recognition would not only make it possible to authenticate the user in different daily activities like: slow walking, normal walking, fast walking even running, but it would also help in avoiding authentication when the user is in passive state like: sitting, standing still, etc. This is one of the key factors that motivated us in making our choice since it is an interesting challenge and it would be benefiting in the data security area if not revolutionized it. This project work has resulted in two publications papers. One paper has been accepted and attached in the appendix. The other paper is submitted and under review.

Acknowledgments

First of all I would like to express my respect and thank my supervisors PhD researcher Mohammad Omar Derawi and Ass. Prof. Patrick Bours, for their support, patience, encouragement and help throughout the completion of my Master's thesis. It is not often that one finds supervisors as Mohammad Omar Derawi who was not only my supervisor, but the best friend for me here in Gjøvik that always finds the time for listening to the little problems and Ass. Prof. Patrick Bours that has always been available with good answers to all my questions and problems. I am truly thankful and I appreciated all the time and advice you gave me. I am really happy to have had them as my supervisors.

Secondly, I would like to thank Prof. Veton Kepuska from Florida Instutue of Technology and Prof. Venet Osmani from the University of Trento for their advices, support and willingness to spend their time and share their own knowledge with me. Next, I want to express my gratitude to all the participants that voluntary participated in my experiments, without them this project would have been impossible. I am grateful to all my good friends for the wonderful friendships and colleges at the NILab. I would really like to thank my friend Zeqir Smajli for his help and support during my studies and Master's thesis, you are a wonderful friend.

Last, but not least, I would like to thank my wonderful parents and my family who even though they were far from me, they supported and encouraged me during my graduate studies. My parents, Shefkije and Naser, receive my deepest gratitude and love for their dedication and the many years of support during my undergraduate studies that provided the foundation for this work. Without you, I would not be where I am today.

Gazmend Bajrami, 30th June 2011

Contents

1 Introduction

Despite the high number of world population, every single person has a unique way of walking due to several different human factors like: aging, injuries, operations on the foot, etc that alter a person's walking style making it slightly different from the others' style. These differences can be either permanent or temporary. A particular way or manner of moving on foot is the definition for gait [1]. According to some studies elders have a reduced range of hip motion at faster walking speeds and 5 degrees less hip extension than in their younger age.

It also appears from early medical studies that there are twenty-four different components to human gait, and that if all the measurements are considered, gait is unique[2] . Scientific studies like those mentioned by Kerrigan and BenAbdelKader has led to a close consideration of gait recognition as an interesting topic and increased the interest in using the way people walk as a form of identifying them. An illustration of the complex biological process considering musculo-skeletal system of human body is shown in the Figure 1. This system can be further divided into numerous types of sub events of human gait. In the same figure are shown the instances used to extract the parameters as an identification system of each individual.

It has been quite a long time that the analysis of biometric gait recognition has been studied [3, 4, 5, 6, 7] with the principal aim: its use in identification, surveillance and forensic systems. Its importance is increasing because of the reliable and efficient means of identity verification it provides.

Today, whenever we use computer systems, they demand authentication as a measure of security. Typically, we perform the authentication at login time with either a password, token, biometric characteristic and/or a combination of these. Performing the last mentioned measure is a stronger guarantee that the claimed user logging in is not a burglar but an authorized user. An issue raises that, not many systems of security requires any further measure once the user is granted access thus assuming that the user is continuously legitimated into the system. Continuous insurance of the user's legitimacy is of high importance in critical or high security environments, this means that it is necessary to ensure that the user is continuously the legitimated one. Therefore, performing the user authentication continuously while the system is actively used is something essential. Nevertheless, this kind of authentication needs to be "attractive" for the user. A very good solution would be something that does not directly imply the user in doing anything special, e.g. periodic password entering. Continuous authentication using biometrics fits these needs. Thus, a choice among the important requirements in continuous authentication is unobtrusiveness, since it is easier to monitor in a non-intrusive way. Relying on current knowledge-based mechanisms the Wearable Sensor (WS) based method can be a very good candidate in fulfilling this requirement.

1.1 Keywords

Gait recognition, Activity recognition, Mobile devices, Wearable sensor, Accelerometer based

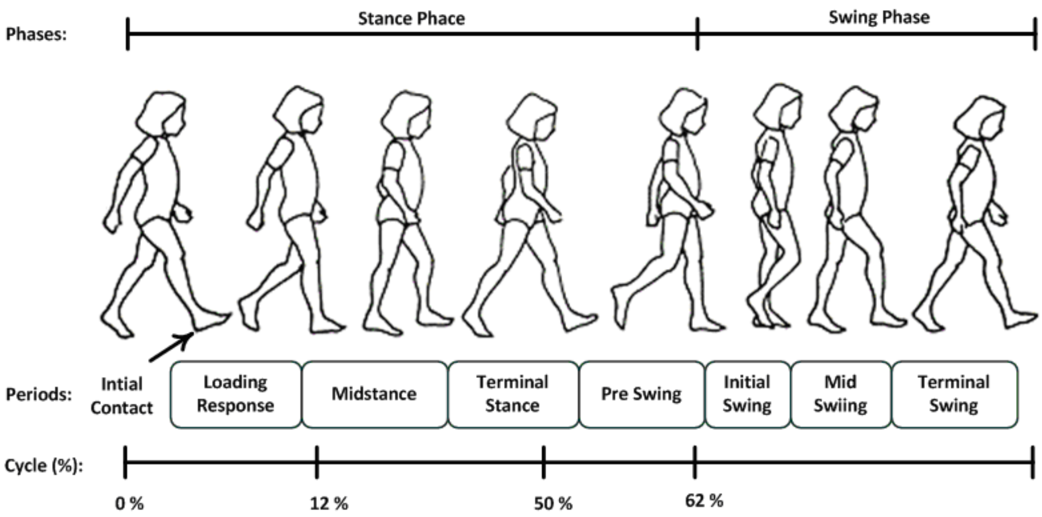

Figure 1: Division of the gait cycle into five stance phase periods and two swing phase periods[8].

1.2 Problem description

An increasing interest has been found in recent research in analyzing human gait as can be read in chapter 3. This project focuses on wearable gait recognition. As far as wearable gait recognition is concerned, the contributions have only focused on the task of person identification on retrieving data from dedicated external sensors and currently is progressing more on gait recognition on mobile phones. This research aims developing and evaluating algorithms for detection of physical activities from data acquired using mobile devices (built-in sensors: accelerometers) that are worn on different parts of the body, which is also known as the

Since wearable gait recognition research has had its focus in extracting feature and evaluating performance, the activity identification has not been developed in wearable gait authentication earlier and especially not regarding the gait recognition with mobile devices. Performing gait authentication under normal everyday walking circumstances is very important in using mobile devices, which will be the first step towards protection mobile devices using the collected acceleration data for security purposes against any unauthorized use of the device and disclosure of information present at the device. It is quite difficult to know whether the authorized subject is performing at a certain time without an activity identifier on the mobile device. As far as the period of performance is concerned the gait recognition should only be functioning when the subject is physically active, and thus, the recognition should not be activated when the subject is passive (sitting down, standing, etc.).

The way this can be reached is by using volunteers to perform experiments and to analyze the data from the experiments. At first the collected acceleration data will be transferred to a computer for further analysis (non-definite sentence)

1.3 Justification, motivation and benefits

Today, accelerometer based gait recognition has been performed using external dedicated sensors and very few papers have been studied using personal devices such as mobile phones. This has lately raised attention world wide. Research within accelerometer based gait recognition have performed experiments with only fixed settings. This means volunteers had to go for a fixed route going back and fourth. Other research papers (which we will have a look at later on in the related work chapter) even performs a little bit more advances walking mechanism, such as normal, fast and slow walking. Very few papers have been working with recognition of activies, especially on commercial devices. Since a mobile phone should serve as a security mechanism, activity recognition is a must to have since to exactly know what the user is doing at a certain time. For example, lets assume a person is walking, thereafter sitting, and at last running. In this case we have three different activities ongoing. The question is then; is it suitable to apply gait recognition while the user is sitting? The phone must know which activies are being performed, since gait recognition should only work when the user is performing any sort of walking, such as fast, slow or normal walking, or even running. Furthermore, it should avoid performing authentication when the user is standing still, sitting, or doing different kinds of abnormal behaviours. This makes activity recognition an important factor in gait recognition using commercial mobile devices.

1.4 Research questions

In this research project the following issues will be addressed:

Technical issues:

- Which methods and techniques can be used to analyze gait data for activity identification?
  In order to analyze the data we collect, they must go through some pre-processing before it can be handed to the actual matching algorithms. Such pre-processing can e.g be noise reduction and time interpolation.
- Has there been any research related work which can be used more or less directly?
  In order to both ease the workload and achieve as good results as possible it is desirable to reuse some algorithms and tools that have already been used in similar topics.

The main questions:

- How can we capture and store acceleration data on commercial available personal devices and locally perform gait authentication on these devices?
- How can we identify different activities (normal, slow, fast) from the captured acceleration?
- How can the use of activity identification be used to increase the performance of gait recognition?

1.5 Planned contributions

The planned contribution of the master thesis for the next year is to find out how activity recognition can be designed most optimaly in accelerometer based gait recognition using mobile phones. We want to make a stable activity identifier, such that we extract the correct walking activies from the full walking signal outputted from the mobile device. Furthermore, we will make look at normal accelerometer gait recognition and use a mobile phone to with a different sampling rate, such that we can reduce the equal error rate by implementing new approaches.

2 Introduction to authentication

This chapter is aimed for those who are relatively new and not so familiar with authentication and biometric. Therefore we will give a brief introduction for these subjects to become more familiar with terms used in the following and more informed about.

2.1 Authentication

Authentication is the process of verifying the identity of a person whether is the person who claimed or declared to be. Authentication is an area which has grown last decades and become most widely used today, by finding application in many places. An important aspect of information security is authentication, which aims to prevent unauthorized access and to decrease the risk against any theft or disclosure of sensitive information. Examples of authentication are passwords which are used to get access to computers, PIN codes are used to get access to our bank accounts and passports are used at border control. The last is an example of human authentication which is used to authenticate or verify a person's identity. We identify friends and family by their voice (when we speak in phone), face, the way they walk, etc. We used both terms authentication and identification above, but there is a difference between these two terms. By identification we mean recognize the identity of a person. Identification is 1: n verification of an identity and authentication is 1: 1 [9]. As we realize there are several ways in which a user may be authenticated, three factors in which authentication methods based on are:

- Something you know (Knowledge based) – For example a secret password, a secret phrase a PIN code etc.
- Something you have (Object based) – For example a smart card, a token, physical key etc.
- Something you are (Biometric based) – For example fingerprint, face recognition, gait recognition etc, a biometric property.

In the following we will describe briefly these three factors.

2.1.1 Something you know

"Something you know" is an authentication method which is based on some secret the user knows and it is the oldest and best known way of identifying oneself [9]. Examples of this method are passwords and PIN codes. Today most popular and widely used method for authenticating users is by username and password. It is most common form to control access to personal computers, networks and Internet. PIN codes are another example of authentication used to get access in bank account and withdraw money form ATM machine. This factor of authentication for a long time it was the only one used, because it is cheap, easy to implement and also very fast authentication method. Due to this, it is used in many different applications. Therefore users often have to used more than one passwords and PIN codes. As result of this for a user is much easier to use the same password/PIN for many different applications or to use easy to remember passwords like family names, birthdays, other important dates, pets and combination of these. This method raises the problem of stealing the password form a user by an attacker by simply guessing it or by using various attacking methods. If users are forced to remember many different passwords or to choose passwords which are difficult to guess this usually leads to the risk that the users will write them down in an easy accessible places or store the passwords in a file and never change. All the drawbacks and difficulties mention above increase the cost of using passwords and PIN codes.

2.1.2 Something you have

In contrast with “Something you know”, in case of “Something you have” the user possesses a unique piece of hardware that can be matched to his identity. Examples of such hardware are keys, tokens, SIM cards, smart cards, bank cards etc. Instead of knowing or remembering longer and difficult passwords which can be forgotten, in this case the user dose no longer needs to remember any password, which is an advantage. The only thing the user needs for authentication is this unique piece of hardware. For an attacker to gain access he must copy or steal the hardware item, which is in most cases very hard to copy and in case of stealing, it can not go unnoticed. The disadvantage of this authentication factor is that not only the hardware items (tokens, smart cards) are expensive, but also the equipment used to verify these items in the verification side. In case of loss or theft of items it is very important to take required action to not be used any longer [9].

2.1.3 Something you are

Since people forget things and lose things, it started growing interest last decade in using the factor “something you are” which is known as biometric for authentication. Most of biometric characteristics are unique to each individual and they are found in almost all people. The chances that two different persons posses same biometric characteristic are very small, even among identical twins. For example fingerprints are unique for each individual. Biometric characteristics include: fingerprint, iris, retina, signature, hand geometry, gait, keystroke, and palm vein, voice, face etc. Due to all difficulties mentioned above for factors “know” and “are”, biometric or factor “are” consider to more robust (against stealing or losing) and an alternative method. For an attacker, depends on which kind of biometric characteristics is used, it can be harder or easier to steal or copy [9]. Biometric characteristics can be classified in two main classes:

Physiological: are the biometric characteristics related to the shape of the part of a human body. Examples are fingerprint, face recognition, DNA, iris and hand recognition.

Behavioural: are the biometrics related to persons behavioural characteristics, such as keystroke recognition, gait-recognition, speech/voice recognition and signature recognition etc.

2.1.4 Two Factor Authentication

Two factor authentication is also called strong authentication means using more the one authentication factors or two identity validation methods instead of one before access can be granted. Combination of authentication factors provides greater levels of security to the systems. These systems are known as multimodal systems. These combination of authentication factors falls into one of four categories:

Know and Have: An example is a personal PIN or password (something the user **Know**) and a security token or Bank card (something the user **has**).

Have and Are: For example a bank card (something the user **has**) in combination with signature (something the user **Are**) instead of using PIN code or a token that needs a fingerprint.

Know and Are: For example using a combination of PIN code (something the user **Know**) with face recognition (something the user **Are**) to access in a laboratory room.

Are and Are: Use combination of multiple biometric modalities, such as using Gait (something the user **Are**) and fingerprint (something the user **Are**) in mobile phone for authentication or fingerprint and face recognition.

When using combination of authentication factors, it is important to make sure that both factors are used and needed for authentication. For example to have access in bank account and make a money transfer we need both the secret password and the token, if one of these misses, we can not first access in make the transfer [9].

2.2 Biometrics: Overview

"Automated recognition of individuals based on their behavioural and biological characteristics."

Biometric identification has been around for a while. Humans have always recognized others through biometrics characteristics such as face, voice and gait etc. The earliest known use of biometric was in 14th century in China where "Chinese merchants were stamping children's palm- and foot prints on the paper with ink in order to distinguish young children from one another"[Moham.]. Shakespeare in his play "The Tempest" made a use of gait "Great Juno comes; I know her by her gait". Alphonse Bertillon did the first scientific literature on 1870's where he describe a system for body measurements for identifying people, which was used until 1920's in USA to identify prisoners. In 1809, Thomas Bewick an ornithologist began to use his fingerprint as his trademark, this is considered to be one of the most important contributions in the study of fingerprint recognition [10]. And in the 1880's Faulds, William Herschel and Sir Francis Galton started working in fingerprint recognition by collecting form criminal sites and manually compared with fingerprints of known criminals [9]. Until 1980's, fingerprints have been the most biometric feature used as a method to authenticate people compare to other biometric features. Around 1980's biometric features such as hand geometry, voice, signature and retina recognition have been used and become popular. Since 1990 commercial face and iris recognition has been around and researches in gait recognition have started last decade. Gait recognition is a relatively new research area.

2.2.1 Biometric characteristics

There are many unique biometric features on humans that can be used for authentication purpose. These biometric features (also called characteristics) can distinguish individuals from each other. Features such as fingerprint, face, and iris and voice recognition are the best known forms of biometrics.
According to [11], for any biometric characteristics to be used for authentication it needs

to satisfy the following properties:

Universality: each person should have the characteristics.

Distinctiveness: Any two persons should be sufficiently different in terms of the characteristics.

Permanence: The characteristics should be sufficiently invariant over a period of time.

Collectability: The characteristics can be measured quantitatively.

In order to be able to use the system, everybody must or should satisfy the first four properties. For a biometric authentication system to be practical, the last three properties should also be considered:

Performance: Measures the recognition accuracy and speed, the resources required to achieve the desired recognition accuracy and speed, as well as the operational and environmental factors that affect the accuracy and speed.

Acceptability: Indicates the extent to which people are willing to accept the use of a particular biometric identifier in their daily lives.

Circumvention: Reflects how easily the system can be fooled using fraudulent methods.

As all these properties requires, a practical biometric system should have the desired recognition accuracy and speed, be accepted by people and harmless, and also should provide proper security against any possible attack.

2.2.2 Biometric systems

Despite the fact that there are many different kinds of biometrics, most biometric systems work in the same way. Biometric systems can operate in two different modes: identification and authentication which is also know as verification. There are two phases in every biometric system: enrollment and verification or identification.

Enrolment: In enrollment phase the characteristics of a user are measured and added to the system. During the enrollment phase, the biometric system transforms the biometric features measured into digital representation and processes these to create a template. A biometric template (also known as template) is a digital reference of information extracted from raw biometric sample. Before the template is created, the quality of biometric features is checked because it is very important to have as high quality as possible. The reason why the quality needs to be high is that it is used every time during the user authentication process or during the second phase of the biometric system. After the features quality is checked, next step is the process of features extraction. Biometric feature extraction is the process of getting relevant information form input data which will be used later during the authentication process. The last step in enrollment phase is the template creation and its store in the system or on an external device such as a Mobile phone, Smart card etc as it is illustrated in Figure 2.

Verification: The second phase is the authentication or verification phase. Most of the steps are the same as during the enrollment see Figure 3. Authentication is the verification process of the unknown user if he/she is the right person who claims to

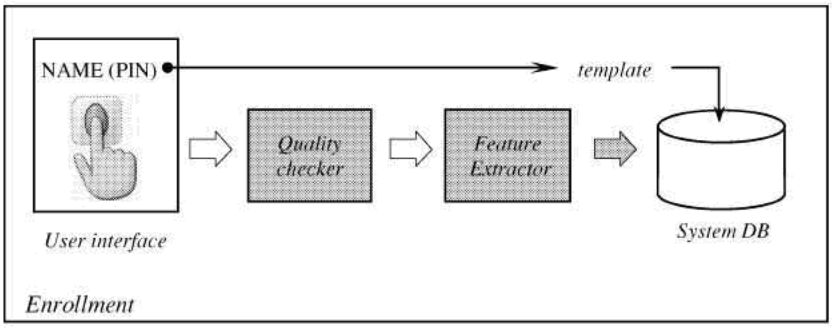

Figure 2: Biometric Subsystems: Enrollment phase[8].

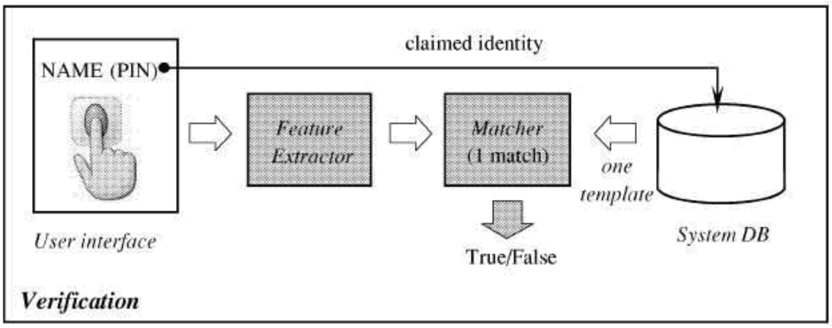

Figure 3: Biometric Subsystems: Authentication phase [8].

be. During the authentication process, the user presents his biometric characteristic to a biometric reader to extract the data that will be used to compare against the template of the claimed identity stored in the database to see if the user matches. In the authentication process the quality of biometric feature captured is not checked like in the enrollment phase, it could in some cases cause rejection of the user because the feature in not good enough. The process of enrollment of user's biometric features in many cases is supervised to control and ensure that the quality of biometric feature is good enough. The new users of biometric devices need to know how to work with these equipment and they need to be supervised during the enrollment in order to get a good quality biometric features.

Identification: When the system is running in the identification mode, the captured biometric from an individual is compare against entire database of templates in order to find a match and identifies the unknown individual and establishes his/her identity. Identification as we mention before is a one-to-many comparison. In an identification system the user does not claim his/her identity. Figure 4 illustrates the identification process.

2.2.3 Biometric system errors

As mentioned before the main advantages of biometrics are that they cannot be stolen or forgotten. Despite of the advantages biometric systems are not perfect. When using passwords and PIN codes, they can be either correct or wrong. But when we use biometric features for authentication they never match 100%. For instance in a fingerprint some of the features extracted are matched with template and some do not match. Various factors

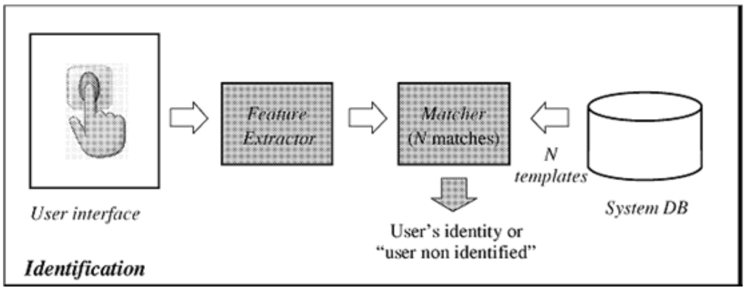

Figure 4: Biometric Subsystems: Identification process[8].

can have impact on captured biometric features such as wet finger print in fingerprint recognition, lighting condition in face recognition, noisy background in voice recognition etc. which makes biometric systems suffer from these problem and which are responsibility for an increase of the number of errors. The similarity between extracted features and the stored template is expressed with matching score. The higher the matching score, the higher the similarity and we are more convinced that this is the right person. The matching score between these two samples is calculated by using distance metrics such as Absolute distance, Euclidean distance and Maximum Difference distance. The calculated distance between two samples is classified as inter-class and intra-class distance. The first one is being used when the distances are measured between two different persons and the second one when the distances are measured between same person. The intra-class gives a low score (distances between same person) and inter-class gives higher score (distances between two different persons). The decision for accepting or rejecting a person depends on the threshold we set for the system. There are three factors on which the performance of the system depends: accuracy, speed and size of the template. The accuracy of a biometric system depends on the number of errors that occurred during the verification process. There are two types of important errors that biometric verification systems make:

- **False Acceptance Rate (FAR)** is calculated from the False Match Rate (FMR). This error happens when a biometric system wrongly accept an imposter user. FAR indicates the number of imposter users which are incorrectly or falsely accepted. property.
- **False Rejection Rate (FRR)** is calculated from the False Non-Match Rate (FNMR). This happens when a biometric system wrongly reject the genuine user. FRR indicates the number of genuine users which are incorrectly rejected.

Other errors that biometric systems can also produce and which should take into account are Failure to Enrol Rate (FTE) and Failure to Capture Rate. According to ISO/IEC JTC 1/SC 37 the definitions for these errors are:

- **Failure to Enrol (FE)** failure to create and store an enrolment data record for an eligible biometric captured subject, in accordance with an enrolment policy
- **Failure to Enrol Rate (FTE)** proportion of biometric enrolment transaction (that did

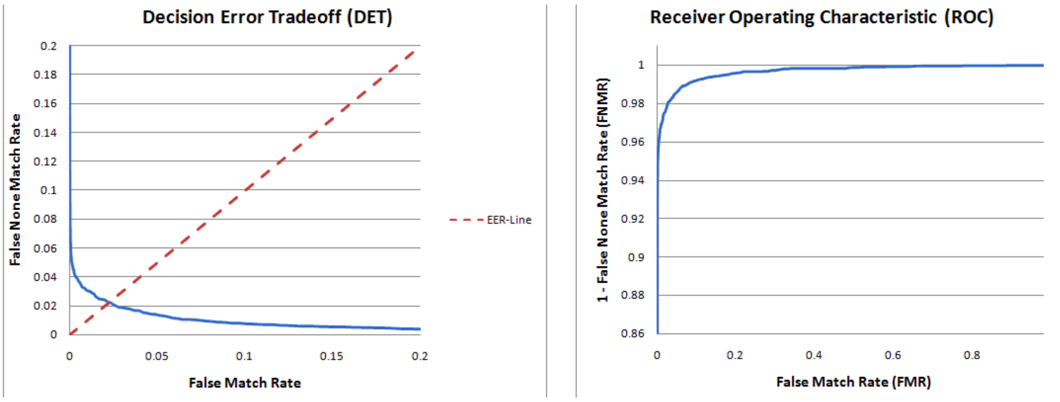

Figure 5: Examples of Decision Error Tradeoff (DET) and Receiver Operating Characteristic (ROC) curves[12].

not fail for non-biometric reasons), that resulted in a FE.

- **Failure to Capture Rate (FTC)** failure of the biometric capture process to produce a captured biometric sample that is acceptable for use (still under definition construction)

The FET increases when the person does not have enough unique features to be captured or the biometric features are not good enough to be extracted and create a biometric feature. The FTC indicates the rate at which biometric capturing device fails to capture the required information when presented correctly [10]. A Receiver Operating Characteristics (ROC) or Decision Error Tradeoff (DET) curve is used to illustrate the error rate tradeoff between FMR and FNMR as it is shown in Figure 5. These curves are used to report the performance of the biometric system.

These two curves show the performance of the biometric systems. Different tradeoff of FMR (or FAR) against FNMR (or FRR) can be produced by changing the threshold value. Therefore it is important to find the value that will produce low FMR and FNMR. The equation for FMR and FNMR are listed below in 2.1 and 2.2

$$\mathrm{FMR} = \frac{\text{Number of accepted imposter attempts}}{\text{Total number of impostor attempts}} \tag{2.1}$$

$$\mathrm{FNMR} = \frac{\text{Number of rejected genuine attempts}}{\text{Total number of genuine attempts}} \tag{2.2}$$

The difference between ROC and DET curves is in the y-axis. DET graph plot the FNMR in the y axis against FMR and ROC graph plot (1-FNMR) in the y axis against FMR. The value of the threshold is assigned depends on the application of biometric system. For application that requires higher security the threshold value should be the one that low the number of FMR/FAR in order to in order to decrease the possibility for imposters to gain access. Another important definition that is used to compare the accuracy of biometric systems is Equal Error Rate (ERR). The lower the ERR the better the systems is. The value of ERR can be obtained from the DET curve by simply drawing an angle of 45 degree line from the (x, y) = (0, 0) as it is shown on the left Figure 5 [12].

2.2.4 Comparison of biometric technologies

It is impossible to choose one biometric feature as the best solution for all situations or to say that this feature is better than another. Each biometric feature has its own strengths and weaknesses. To decide which feature to use depends on the situation and the user demand. A way to classify biometrics is by using biometric characteristics which are described before. Table 1 shows the comparison between different biometrics.

Table 1: Comparison of Various Biometric Features [9]

Biometric Features	**Univ**	**Dist**	**Perm**	**Coll**	**Perf**	**Acce**	**Circ**
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	H
Face	H	L	M	H	L	H	H
Facial Thermogram	H	H	L	H	M	H	L
Fingerprint	M	H	H	M	H	M	M
Gait	M	L	L	H	L	H	M
Hand Geometry	M	M	M	H	M	M	M
Hand Vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	L	M	M
Odor	H	H	H	L	L	M	L
Palmprint	M	H	H	M	H	M	M
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

3 Related work

This section is divided into two subsections. First subsection describes the motion-based (gait biometrics) identity verification. Second subsection introduces activity recognition.

3.1 Gait Recognition

Despite several different categories of gait recognition, it is categorized in three basic approaches of identity verification based on gait, in particular approaches where motion information is acquired by

- video cameras,
- sensors installed on the floor and
- wearable sensors attached to various location on the body (clothes) of the user.

Our focus mainly stands on WS-based approach and on illustration of this approach, in general and how it nicely meets the required continuous authentication in particular, that were previously specified. In the chapter are also discussed the best possible body locations where motion-recording sensors (MRS) could be attached or worn. Some examples are also provided regarding the performance accuracies of such locations. There are three different approaches in gait recognition; Video Sensor Based (VS), Floor Sensor based (FS) and Wearable Sensor based (WS).

3.1.1 Video Sensor Based

The system of video sensor approach would typically consist of several digital or analog cameras (black-and-white or color), with suitable optics in order to acquire the necessary gait data. With the use of techniques like thresholding to convert the images into simply black and white; background segmentation, which performs a simple background subtraction or pixel counting to count the number of light or dark pixels; could be several possible ways in identifying a person. Figure 6 shows an example of the VS-based approach with processed background segmentation.

Figure 6: Background segmentation for extracting the silhouette picture (subtraction)[13].

During the last decade until currently a scientist has been working in doing the analyze of the movements of criminals caught on CCTV in order to compare them with those of a suspect [14]. In December 2004, there was a case where a perpetrator robbed a bank in Denmark[3]. Two surveillance cameras were recording all the process of robbery. One camera placed at the entrance that recorded the robber's frontal view (process of walking in, standing and walking in the bank during the robbery, and leaving the bank). The other camera placed inside the bank that recorded the cashier's desk, provided the persecution enough evidence to convince the court rely on the gait-analysis tool to find the perpetrator of the robbery. In about the same time in late December 2004, there was a murder crime scene in the United Kingdom. A podiatrist explained the supreme court that the person captured on the video and he some other previous videos of the murderer matched [15]. The other case occurred around mid-April 2008, when a burglar was caught because of his bow-legged walk [16]. Despite the fact that the burglars face was unable to be seen, they could identify the burglar. Even though in most cases during the robbery, the perpetrator wears a mask to hide his body characteristics of identity such as face and hands so no evidence like face or fingerprints could be shown or found, still cameras are available and useful in recording the gait where enough information can be used in the process of perpetrator identification.

3.1.2 Floor Sensor Based

The floor sensor approach, considers placing the censors along the floor (on a mat) where gait data can be measured while people walk across. The FS-based differs from the WS-based since it is the force to the ground by humans walk to be considered, that is also known as the GRF (Ground Reaction Force). In a research from the University of Southampton, such a floor sensor for gait recognition was prototyped as illustrated in Figure 7.

3.1.3 Wearable-Sensor Based

Apart from the video sensor (MV) based and floor sensor (FS) based gait recognition, another gait recognition approach appeared recently and that is wearable sensor based. It includes relying on attaching or wearing motion recording sensors on the human's body in different places like; on the pockets, waist, shoes and so forth. These sensors (WS) can have several purposes due to retrieving numerous types of data. This means that sensors of different types can be used for instance gyro sensors (measure rotation), accelerometers (measures acceleration), force sensor (measures the force when walking) etc, so far a great focus has been shown on accelerometer based gait recognition. Thus, these accelerometers are becoming an important tool into our every-day life. Some of the newer mobile phones nowadays, e.g. the iPhone, are already using the wearable-sensors; they use built-in accelerometers in order to detect when the device rotates, so it can tell whether to display what is on the screen in vertical or horizontal format. This gives the user a better view which format is best for viewing, such as a photo, web page, video. Moreover, these devices can further be used in detecting when it is being lifted to the ear so that phone calls are answered automatically. All these successful practices of gait recognition sensors in modern technology have increased the interests in researching at different methodologies to analyzing the features especially in wearable-based gait biometrics. However, feature extraction from gait signals is crucial for the efficient gait recognition. For a general gait analysis, the signal processing flow is shown in Figure 8.

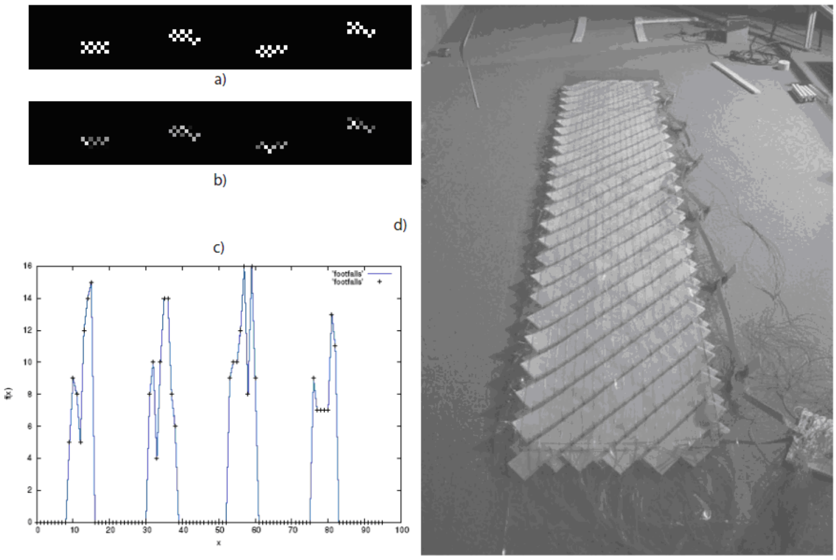

Figure 7: Gait collection by floor sensors. a) shows footsteps recognized, b) shows the time spent at each location in a), c) shows footstep profiles for heel and toe strikes (x and f(x) indicate the heal/toe locations and footfalls forces, respectively), and finally d) is a picture of a prototype floor sensor carpet[17].

A WS-based gait recognition application could make a significant improvement of the authentication in electronic devices. Among the advantages of WS-based gait recognition and the main argument towards CA is its unobtrusiveness. An example would be the integration of the MRS with clothing (e.g. footwear) or personal electronics of the user.

While the user is waking the motion is recorded by MRS so the recording could be used to verify the identity of the user in a specific background application. Since the data would be collected continuously they could be of a good use for continuous identity verification in mobile phones as well. The continuous identity verification can ensure that the same authorized user is using the phone in every step he/she performs and not someone else. The recording equipment like MRS is quite cheap so many personal electronic equipments (e.g. mobile phone) are equipped with similar sensors recently.

Experiments

Regarding the experiments made so far there are no public database created for accelerometer based gait recognition although the researchers have created some experiments and databases of their own. In Table 2 we have a summary of the performed experiments in the research including the environment of the activity performed, the type of the activity and of course the the range of walking for each subject.

The above mentioned experiments are all controlled experiments except [19]. A controlled experiment is a fixed laboratory setting which means it is quite different from a real world scenario due to its importance in getting as more exact results as possible during the research. While in everyday life people keep their mobile phone in their pockets or hold it time after time, the phone continuously moves in different directions, it rotates and is of a much better use in a fixed setting we usually attach it to a single part of the body during the whole time. As shown on the Table 2 the number of volunteers

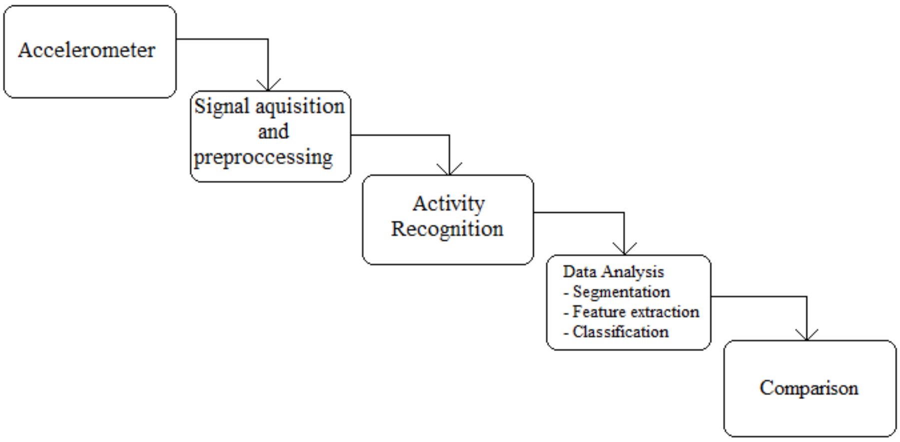

Figure 8: Processing flow of method for gait verification.

Table 2: Experiments Summary

Study	Walking activities	Subjects
[18]	treadmil(normal, fast, slow)	5
[19]	free normal, free resting	5
[20]	normal, fast, slow	6
[21]	normal	20
[22, 23]	normal	21, 35
[24]	normal	21, 30, 50, 100
[25]	normal	36
[10]	normal, fast, slow, circle	60

differs quite a lot. The number of test-subject has been low so far and it has resulted in providing dissimilar performance. The different number of volunteers makes the recognition performances be incomparable. Clothing appears to be another issue because gait is different from person to another and clothing may turn out to be a critical parameter in affecting the gait-recognition research outcomes. Moreover, only a few studies have resulted in different behavioral settings and a study has shown there is a slight change of the gait-signal of one person one day to another[10].

Data acquisition

There are several types of equipments available to gain the accelerometer data: a dedicated accelerometer, GPS device, mobile phone, etc. these accelerometers measure the acceleration of three directions, first up-down or x-direction, second forward-backward or y-direction and third sideways or z-direction. An overview of the placement of sensors and their models used in the literature is given in Table 3 [26] .

Depending on where the accelerometers are built (into cell phones or dedicated devices) they normally output different sample-rates per time unit. Most accelerometers basically have a low sample-rate/frequency whereas only a few have a high frequency rate. Furthermore, a considerate number of devices nowadays contain multiple sensors, like gyroscope, magnetic-field etc.

Table 3: Data Acquisition Summary

Study	Acquisition Form	Subjects
[27]	shoe	MEMS accelerometer
[28]	breast/hip	cell phone accelerometer
[29]	whole body weight	force plate
[24]	ankle/pocket/arm/hip	3D accelerometer (MRS)
[22, 23]	waist	3D accelerometer (analog)
[30]	leg	wireless accelerometer(Tmote Sky)
[19]	pockets	phone headset
[31, 25]	waist	3D accelerometer (ADXL202JQ, analog)
[20]	hip	cell phone accelerometer
[18]	ankle	3D accelerometer
[21]	elastic belt on body	3D accelerometer
[13, 10]	hip	3D accelerometer (MRS)

Preprocessing

Different performance of preprocessing has been made in literature and it hasn't been performed in every study. The measured signals of acceleration are sometimes components of low-frequency. These outputted signals are easily affected by different environmental noise of the experiment like the equipment's electronic noise the high frequency noise, etc that is quite likely to obscure clarity of the acceleration data. Table 4 overviews preprocessing methods applied.

Table 4: Preprocessing Approaches

Study	Type	Approach
[24]	Time interpolation	Linear time interpolation
[13, 10]	Noise filter	Weighted moving average
[10]	Noise filter	Moving Average
[23, 29]	Noise filter	Daubeshies wavelet (wavelet transform)

Data Analysis

User identification from gait patterns with accelerometers used is based on the assumption of the gait acceleration profile being unique at some extent for each individual. First, it is important to compute the feature template vector (that represents the characteristics of the person's gait) to authenticate and of course to store it as a template. This feature vector is computed during the process of authentication and compared to the feature template. An effective analyze of the accelerometer data can be made in two domains; Time-domain and/or the frequency-domain. The aim of the time-domain is analyzing the three acceleration signals (x,y,z) and monitoring how these three signals change over time (t), whereas the aim of the frequency domain analyze is showing how each frequency band of frequencies is given. A given function or signal can be converted between the domains of time and frequency by using some mathematical operators known as transformation. Therefore, the researchers are the ones to decide which of the two abovementioned domains, they will use. Another possibility would be finding a way to somehow combine both domains.

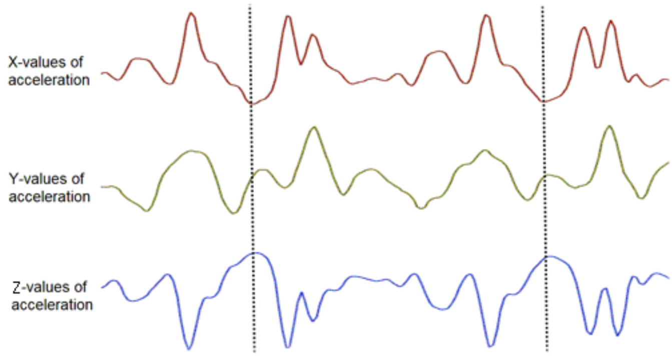

Figure 9: One gait cycle: begins when one foot touches the ground and ends when that same foot touches the ground again.

Segmentation

The process of identifying boundaries in gait signals is known as data analysis or segmentation. It can be performed in different ways and is a very important sub-problem. The signals gained from different individuals are a composition of periodic segments recognized as gait cycles and they correspond physically to two alternative steps of the individuals. These cycles begin as soon as the foot touches the ground and finishes when the same foot touches the ground for the second time, this process is shown in Figure 9.

The process is repeated with the beginning of the third step that performs the end of one gait cycle is the beginning of the other. In order to split the signal into gait cycles, we need a determination of the gait cycle periods and it can be performed in different ways. One way could be by using the x, y and z data in separate ways or another way could include a combination of two or three of the axes data. A summary of the three segmentation approaches applied so far is shown in Table 5 [26].

Table 5: Segmentation Approaches

Study	Approach
[30]	Period of an periodic gait cycle
[31, 25]	Cycle Detection Algorithm (1 step extraction)
[24, 10]	Cycle Detection Algorithm (2 step extraction)

Feature extraction in the time domain

The time domain is the analysis of signals, with respect to time as mentioned earlier. One of the first applied methods in gait biometrics is the average cycle method and has been the most applied methods so far. This method is a simple approach which obtains the average of all extracted cycles. Despite the frequent use of average cycle method some other extraction approaches have been successfully developed as well. Table 6 shows the extractions developed until late [26].

Feature extraction in the frequency domain

The extraction of the features in the frequency domain differs slightly from the time domain, as other (mathematical) approaches need to be applied. Among the most efficient

Table 6: Time Domain Feature Approaches

Study	Approach
[13]	Matrix with cycles
[31]	Average cycle detection
[25]	N-bin normalized histogram
[20]	Cumulants of different orders

ones is known to be the fourier transform. It is a mathematical operation which makes a transformation of the signal from the time domain to the frequency domain, and vice versa. The overview of some other applied methods is shown in Table 7 [26] .

Table 7: Frequency Domain Feature Approaches

Study	Approach
[23]	Discrete Fourier Transform (DFT)
[18]	Fast Fourier Transform (FFT)
[32]	Discrete Cosine Transform (DCT)
[29]	Discrete Wavelet Transform (DWT)
[28]	Wavelet Packet Decomposition (WPD)

Comparison functions

To compare two feature vectors with each others we apply a comparison function, for instance the use of the distance metric function would be an appropriate comparing function. In mathematics, the distance metric function is a process of defining a distance between several elements within a set. The numbers of distance developed functions reach the infinite. Depending on the metric, distance functions give quite different results. This has a major impact in authenticating therefore the importance lays in finding or creating as suitable metric as possible. Finding out the similarities of one individual to another is very important in biometrics. The comparison functions that have been used are shown in Table 8 [26].

Table 8: Comparison Approaches

Study	Comparison Metric
[31]	Cross-correlation
[10]	Euclidean Distance
[24]	Absoulte (Manhattan) Distance
[13]	Dynamic time warping (DTW)

Classification

An important well-studied area used within gait recognition is also the (un)-supervised learning approach. The supervised learning in wearable gait recognition is a sort of machine learning approach used to get deductive measures of a function derived from gait signal training data. The training data consist of data extracted from the accelerometer signals attached to the equipment. The function output should be a value extracted continuously or it can predict a class label of the input known as classification. Table 9 shows an overview [26].

Table 9: Classification Approaches

Study	Comparison Metric
[20]	Support Vector Machine (SVM)
[20]	Principal Component Analysis (PCA)
[30]	Linear Discriminant Analusis (LDA)
[20]	Multilayer perceptrons-neural network
[28]	Kohonen self-organizing map (KSOM)

The purpose of data analyses is creating a template to represent the subject. Exploring accelerometer based gait recognition started in 2005, and it has resulted in some different data analysis methods like Average Cycle Method (ACM). The ACM has increased its popularity because of the simplicity it offers as a feature extraction method on creating templates. Other different features that have been used in creating templates and comparing are correlation, cumulants, histogram similarity, ACM, FFT coefficients, and other regular features. Estimating if some of the mentioned techniques are generally practical for data from different equipments is quite difficult due to the variation of a large extent of performed experiments and the applied analyses.

Comparing gait performances

On wearable gait there is no public data-set available for a difference from video-based gait biometric. And this complicates the comparison issue in comparing multiple private-sets with each other. So, we can not consider any direct comparison in this section. Nevertheless, all results will continuously be overviewed. A short summary of the current WS-based gait recognition studies from years 2004 to 2010 is shown in Table 10 [26]. In the last column, #TP, is represented the number of test-persons that were a part of the research.

Table 10: Performances of current wearable sensor-based gait recognitions

Study	EER	Recognition	TP
[33]	-	96.93 %	9
[34]	-	97.4 %	10
[23]	5.6 %	-	21
[24]	5 %	-	30
[35]	13.7 %	-	31
[31]	6.4 %	-	36
[25]	7.0 % , 19.0 %	-	36
[36]	1.68%	-	60
[13]	5.7%	-	60
[10]	5.9%	-	60

3.2 Activity Recognition

Activity recognition is the process of identifying everyday common human activities in real life. It is a new area of study, and is becoming an interesting research field due to different areas of application. Accelerometers come integrated on new models of mobile devices such as smart phones, tablet computers, digital audio players (Ipod) etc., which record the body motion. The majority of studies for activity recognition are performed by

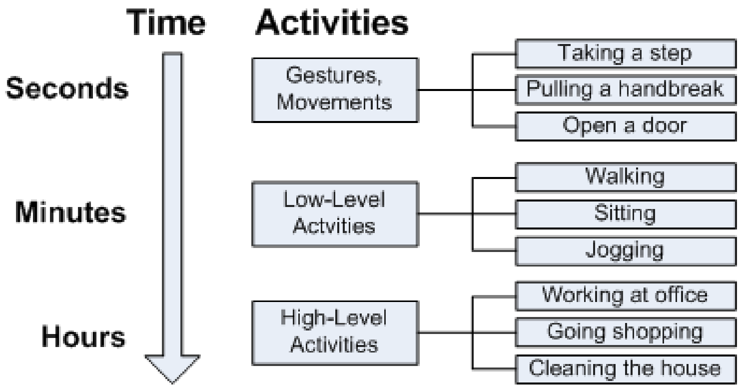

Figure 10: Level of Activitities [37].

using wearable sensors. Several studies have shown that wearable sensors are adequate for activity recognition. In the following we will show some of the sensors that have been used so far for activity recognition, a summary of different activities that were recognized by using various sensors and the approaches used for identifying different human activities.

Due to many different application areas of activity recognition, there is no surprise that the list of activities that many researchers have tried to recognize with various sensors is long.
According to [37], activities can be categorized in three groups based on duration and/or complexity: *Gestures (or Movement/Motif)*, *Low-Level Activities*, and *High-Level Activities*. Activities such as walking, sitting, standing, eating, cleaning windows are considered as low-level activities which usually last between seconds and several minutes. As high-level activities are considered activities like sightseeing, cleaning the house, working at office, that usually last for more than a few minutes up to a few hours. Figure 10 illustrates these groups of activities.

3.2.1 Experiments (Activities)

The identification of everyday routine and leisure activities such as walking, running, biking, sitting, climbing and lying have already been analyzed in laboratory settings by several researchers. All these studies were done by different sensors such as accelerometers which were embedded in wearable sensing devices to collect the needed data. The types of sensors used for activity recognition are to be discussed in the next section. Accelerometer sensors are very useful for low-powered equipments like smart phones, tablet computers with applications that are suitable for real-time detection of user's activities. Physical activities such as walking, walking up/down stairs standing, sitting, and running have been studied by some of the researchers using different accelerometers sensors. Table 11 summaries different activities by different studies.

Another class of activities, mainly studied in healthcare environments, are the so-called "Activities of Daily Living" (ADLs). ADLs include activities such like bathing, toileting, dressing, feeding ourselves, homemaking which are basic skills needed for daily self-care activities. A set of ADLs is known as the "Instrumental Activities of Daily Living" (IADLs), those are skills beyond basic self-care which a person needs to perform for an independent living. IADLs include activities like shopping, driving, cleaning, cooking,

Table 11: Activity recognition research studies. #TP = Test Persons

Study	Activities	#TP
[38]	walking flat, walking slope-up, slope-down, walking stairs	52
[39]	sitting, walking, jogging, walking stairs, standing	29
[40]	sitting, standing, and walking	26
[41]	walking, running, cycling	24
[30]	walking, running, sitting, standing, bicycling	20
[32]	walking, climbing stairs	15
[4]	lying down, sitting and standing, walking, running,	12
[42]	sitting, standing, walking, walking stairs, riding elevator up/-down, and brushing teeth	12
[43]	running, still, jumping and walking	11
[44]	sitting, walking, walking (street), waiting at a tram stop, riding a tram	8
[14]	walking, standing, sitting and running, walking stairs	6
[45]	sitting, walking, running, walking stairs	6
[46]	standing, walking, running, climbing	5
[47]	standing, sitting, lying, walking, running	5
[48]	sitting, walking, jogging, riding a bike, walking stairs	2

doing laundry and managing money. Table 12 shows an overview of these activities.

Table 12: Studies of activity recognition of daily living (ADL)

Study	Activities (ADL)	#TP
[49]	toileting, washing, housework, leisure activity, oral hygiene, heating use, taking medication, etc.	14
[50]	mopping, cleaning windows, making bed, watering plants, washing dishes, setting the table, vacuuming, ironing, dusting	12
[51]	lying, rowing, cycling (training,regular), sitting, standing, running, walking, football	12
[52]	prepare food, clean dishes, wash clothes	10
[53]	showering, urination, flushing, washing Hands, defecation, brushing teeth	4
[54]	prepare food, toileting, bathing, dressing, grooming, preparing a beverage, doing laundry, etc.	2
[55]	prepare different food, eat cereal, dust, brush teeth, tend plants, set table, clean windows, take medication, shower, shave	2

3.2.2 Data Acquisition

Depending on the activities there have been used several kinds of sensors in the data acquisition process for activity recognition. As mentioned earlier, accelerometer sensors are adequate and most commonly used for continues activity recognition. They are also considered to be less intrusive than other sensors such as RFID gloves, microphones, and cameras [37]. Therefore, accelerometers are becoming very important tools due to many advantages in activity recognition. There is not a single sensor that can record all the body movements and recognize all kind of human everyday activities at one time. Therefore, most researches today have been using different sensors to capture the data and multiple sensors attached on multiple parts of the body such as, hip, wrist, arm, ankle, chest, thigh, knee. For instance, activities like walking fast, walking slow, and running can be recognized by motion sensors but these sensors can not recognize activities such as, talking, reading, driving car etc.. Table 13 overviews some of the most widely used sensors for activity recognition research.

Table 13: Sensors used in different studies.

Study	Sensor Placement	Sensor
[56]	Above ankle, above knee,hip, wrist,elbow,	3D Accelerometer (ADXL311)
[57]	Belt (left/right)	3D Accelerometer ADXL202
[58]	Chest	3D Accelerometers (ADXL213, analog)
[59]	Hip, thigh, ankle, arm, wrist	2D Accelerometer (ADXL210E, analog)
[60]	Legs	2D accelerometer (ADXL202JE, analog) and Ball Switches
[61]	Legs (upper), above knee	1D Accelerometer (ADXL05s, analog) , passive infrared sensors, carbon monoxide sensor, microphones, pressure sensors, temperature sensors, touch-sensors and light-sensors
[62]	Near pelvic region	3D Accelerometer (CDXL04M3)
[43]	Pocket	3D Accelerometer (ADXL330, analog)
[39]	Pocket	3D Accelerometer (Cell phone)
[63]	Pocket	2D Accelerometer (ADXL202), GPS
[64]	Shoulder	Sociometer (IR transceiver, a microphone, two accelerometers, on-board storage, and power supply)
[65]	Waist	3D Accelerometer
[66]	Waist	3D Accelerometer and a microphone.
[67]	Waist belt	3D Accelerometer
[68]	Wrist, hip and thigh	2D accelerometer (ADXL202JE), Tilt switches

Other sensors that have been used for activity recognition are: GPS sensors [51], vision sensors (i.e., cameras) [51, 69], microphones [53, 70], RFID tag readers [71, 49, 50], ball switches [60], fibber optical sensors [72], gyroscope [73], body and skin temperature sensors [61, 74, 75, 76, 4], light sensors [61, 74, 77, 78], foam pressure sensors [79], pressure sensors [74], physiological sensors [80], humidity and barometric sensors [74].

3.2.3 Activity Recognition Process

Segmentation

Detection of activities from the collected data is the process of finding the "boundaries" for different activities in the accelerometer signal. Segmentation is a necessary step in the data analysis process before the feature extraction and the classification. Several segmentation techniques have been used to identify different activities from the sensor data. Some of the segmentation methods that have been used for activity recognition are: "Sliding Windows", "Top-Down", "Bottom-Up" and "Sliding Window and Bottom-Up (SWAB)" [81].

Feature Extraction

The input data recorded with the sensors from the human body motions is too large for processing, thus it is easier as an initial step to transform the large input data into a reduced representation set of features before further processing. The process of transforming the large input data into the set of features is called feature extraction. The feature extraction is a very important step; therefore features should be carefully chosen in order to extract relevant information from the input data, because it will have a strong influence in the results of classification. Features selection is an important and essential step in the design of any activity recognition system, in order to design an effective system. The features in different studies were analyzed mainly in time-domain and frequency-domain. In the following we will brief describe features extraction in the time-domain and frequency-domain.

Feature extraction in the Time-Domain

In much of the research, studies were considering only time-domain features due to avoid the complexity of pre-processing that required transformation of the signal into frequencies. They consume little processing power and the algorithms can be applied directly. Table 14 shows a summary of papers that consider the time domain features.

Table 14: Feature extraction studies in the time domain

Study	Approaches
[82, 59, 83, 56, 62, 84, 85, 86, 77, 87, 47, 88, 89, 55, 50, 90, 48]	Mean
[56, 88, 62, 82, 66, 87, 84, 85, 86, 77, 55, 50, 90, 48]	Variance or standard deviation
[85, 77, 84, 89]	Root mean square (RMS)
[89, 77, 82, 87, 78]	Zero or Mean Crossing Rate
[87, 85, 61, 89]	Derivative
[60, 91, 92, 93, 94]	Peak Count and Amplitude

Feature extraction in the Frequency-Domain

Unlike the time-domain features, the signal should transform data into the frequency domain and this process requires pre-processing and different transformations such as the use of Fast Fourier transform (FFT). Table 15 shows the most widely used features in frequency domain.

Table 15: Feature extraction studies in the frequency domain

Study	Approaches
[75, 61, 55, 50, 90, 70, 48, 86, 42, 95]	Fast Fourier Transform
[55, 50, 90, 62, 86, 42]	Energy
[55, 50, 90, 48, 96, 86, 42]	Spectral Entropy
[73, 96]	Frequency range power

Classification

Next step after the feature extraction is the classification process. In the classification process, the classification algorithm builds up a model (classifiers) for different human

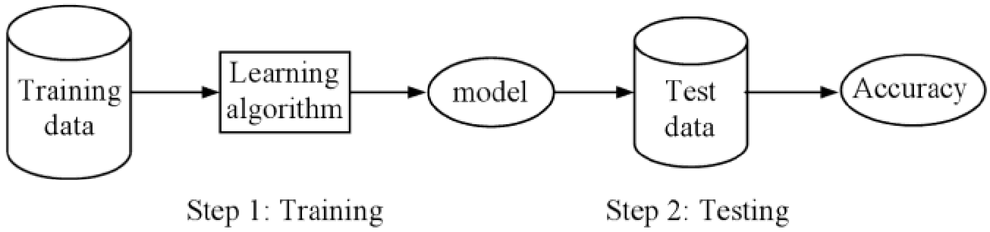

Figure 11: The basic of learning process: training and testing [97].

activities and then uses these classifier to identify human activities from the test data. A wide range of machine learning approaches and algorithms are used for activity recognition. Most of these approaches have been used for activity recognition which can be categorized into two groups: supervised learning and unsupervised learning.

Supervised learning is a machine learning technique, also sometimes called "learning with a teacher" in which the system is trained by using a set of training data before it comes into use in classifying the test data. There are two general phases in a supervised learning technique: training and testing. During the training phase the system is taught (trained) by using a set of training data to create a classification model to classify unknown data. During the testing phase, the model of the system is tested using a set of test data to measure the classification accuracy [97]. Training and testing phases are illustrated in Figure 11.

The majority of works in activity recognition have been done by using supervised learning methods. A summary of these approaches applied so far is shown in Table 16. Supervised learning techniques are mostly used for activity recognition in majority of the researches. Next step is to look at the unsupervised learning techniques which are dissimilar than the supervised learning.

Table 16: Supervised learning approaches used for activity recognition

Study	**Approaches**
[94, 59, 98, 62]	Naive Bayes Classifier
[59, 77, 62]	C4.5 Decision Tree
[73, 68, 61, 62]	Nearest Neighbor
[93, 48, 71]	Hidden Markov Model
[68, 62]	Support Vector Machine
[61]	Kohonen Self-Organising Map

Unsupervised learning by contrast does not use any training or testing data. Instead, it "tries" to classify the unknown data by separating the data into different classes (clusters). It is a "learning without teacher" method. The method tries to directly build models not basing itself on any priori-built model or knowledge. It learns from the unlabeled data, the task of this method is to discover classes of similar examples from the unlabeled data and organizes data into similarity groups, which is known as clustering, or by estimating the distribution of data within the input space which is called density estimation [99]. Clustering is the process of organizing unlabeled data into clusters, where the data in the same cluster are similar to each other and the data in different clusters are dissimilar

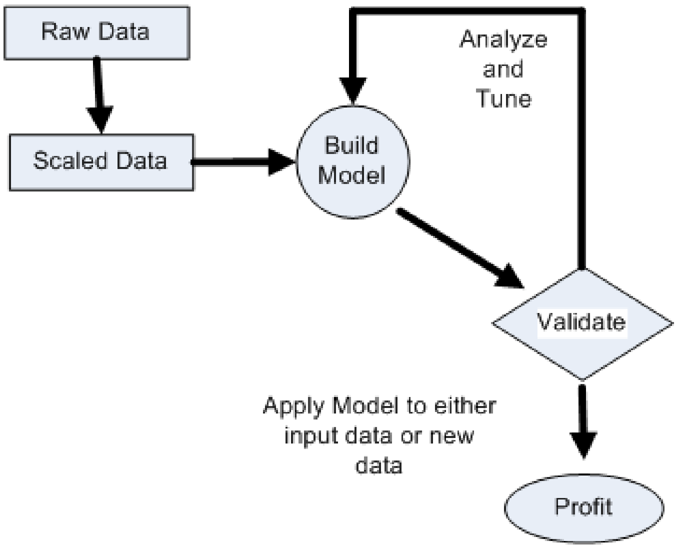

Figure 12: Unsupervised Learning Workflow [107].

[100]. A summary of the unsupervised learning approaches that are applied for activity recognition is shown in Table 17.

Table 17: Unupervised learning approaches used for activity recognition

Study	Approaches
[101, 102, 103]	Hidden Markov Model (HMM)
[104]	Hierarchies of HMM
[105]	Hierarchical Dynamic Bayesian Network
[22]	Multiple Eigenspaces
[103]	Gaussian Mixture Models
[106]	Multi-layered FSM

The process flow for unsupervised learning is illustrated in Figure 12.

3.2.4 Activity Recognition Performances

Studies have shown different accuracies for activity recognition systems in which the data collection was performed in a controlled laboratory settings (subjects are told how to walk, run etc.), from the experiments in which the data was collected under normal circumstances. As we saw in the data collection section a range of different sensors are used to collect the data. Experiments were performed by placing these sensors in one or multiple locations on the body. A summary of recognition accuracies is shown in the Table 18.

Table 18: Recognition Accuracies.

Study	Recognition Accuracy	Activities Recognized	#TP
[41]	80%	walking, running, cycling, driving, sports	24
[59]	84%	walking, sitting, standing, running, computer work, bicycling, Lying down, etc.	20
[57]	83% - 90%	walking, downstairs, upstairs, opening doors	6
[108]	90%	walking, jogging, upstairs, downstairs, sitting, standing	29
[109]	90.8%	walking (slow, normal, fast), sitting, standing, lying, falling	6
[92]	92.85% - 95.91%	sitting, standing, walking,	8
[56]	65% - 95%	sitting, standing, walking, stairs up/down, whiteboard writing, shake hands, keyboard typing	1
[43]	97,51%	walking, jumping, still, running	11
[47]	99,5%	standing, sitting, lying, walking, running	5

4 Experiment

This chapter describes the technology used to perform the experiment for data collection and a detailed explanation for the execution of the experiment.

4.1 Technology

In order to acquire acceleration data we have used a Mobile Phone called Motorola Milestone. It consists of a triaxial accelerometer that enables measuring body motion, magnetic field sensors, a temperature sensor, a proximity sensor, etc. This is a new generation of smart phones with Android Operating System, which is equipped SD card on which the acceleration data are stored, a Bluetooth and USB connector that makes it possible to transfer the data form mobile phone to a computer. An LIS331DLH 3-axis accelerometer sensor is integrated in the Motorola Milestone phone. The acceleration range of the accelerometer is between -2g and +2g with a frequency sampling about 100 samples per second. The mobile phone we used in our experiment is shown in the Figure 13.

In order to read the data from the accelerometer we developed a program in Java called "*DataCollctotToll*". The data collected from the accelerometer by this application tool are values in x, y, z direction and the time as it is shown in Listing 4.1. In Figure 14(a) is illustrated how a cycle is represented in X and Y direction and the resultant. In Figure 14(b) we can see the foot movement. The Z values are excluded from the Figure 14 as these values present the sideways movement and do not have cyclic repetition like X and Y. When the application is started, by pressing the "Start" button a text file named *AccValues Mar_1,_2011_1.05.23_PM.txt* is created and stored on the SD card. Every time the application is started an existing file with that name is renamed by changing date and time in the file. This means that we do not need to worry about accidently overwriting existing data.

Listing 4.1: Output data from the mobile device with the time, X, Y and Z values, respectively

```
   Time          X           Y           Z
8545773985  0.26477954  1.1375713   9.414384
8545781431  0.26477954  0.9610517   9.365351
8545789152  0.26477954  0.69627213  9.365351
8545796995  0.46091253  0.69627213  9.522257
8545804746  0.46091253  0.6276256   9.698776
8545812895  0.4118793   0.6276256   9.885103
8545821256  0.24516624  0.57859236  9.95375
8545828581  0.28439283  0.57859236  10.012589
8545836637  0.28439283  0.57859236  10.071429
8545852293  0.2353596   0.57859236  10.228335
8545860166  0.2353596   0.75511205  10.149882
8545867582  0.2353596   0.75511205  10.071429
```

Figure 13: Motorola milestone, a smartphone with a builtin triaxial accelerometer.

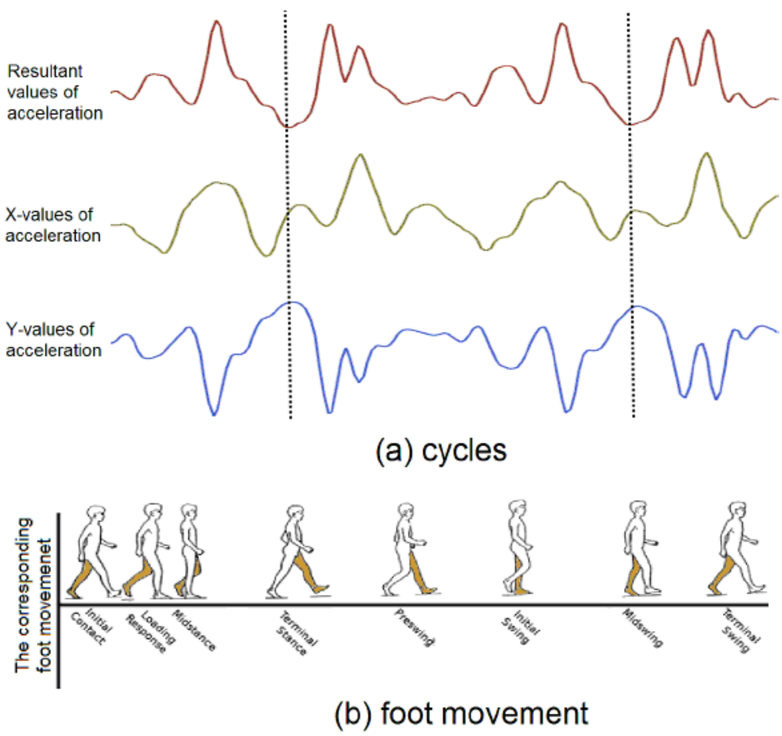

Figure 14: (a) An illustration showing the cyclic repeating of gait cycles in R, X and Y acceleration. (b) The actual foot movement[10].

	Participants	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk	Walk
1		Norm	Fast	Slow	Norm	Fast	Slow	Slow	Norm	Slow	Fast	Norm	Slow	Norm	Fast	Fast
2		Norm	Fast	Slow	Norm	Fast	Slow	Norm	Slow	Norm	Fast	Slow	Fast	Fast	Norm	Slow
3		Norm	Fast	Slow	Norm	Fast	Slow	Norm	Fast	Norm	Slow	Norm	Slow	Fast	Slow	Fast
4		Norm	Fast	Slow	Norm	Fast	Slow	Slow	Norm	Fast	Norm	Slow	Fast	Fast	Slow	Norm
5		Norm	Fast	Slow	Norm	Fast	Slow	Fast	Slow	Slow	Norm	Fast	Slow	Norm	Fast	Norm
6		Norm	Fast	Slow	Norm	Fast	Slow	Fast	Slow	Norm	Slow	Norm	Fast	Fast	Slow	Fast

Figure 15: Experiment Execution: the 6 first walks were fixed and the last 9 walks were randomly chosen.

4.2 Experiment details

4.2.1 Main experiment setup

To answer the research questions raised in this master project, we performed an experiment. In the experiment we asked the volunteers to perform different activities in order to look if we are able to recognize them. These activities include:

Normal walk: The participants were instructed to walk at a normal velocity.

Fast walk: The participants were instructed to walk faster than their normal walking velocity. It is important that the participants walk significantly faster than in normal walking, in order to be able to distinguish these two different activities, but it is also important during fast walking that at any point at least one foot of the subject to have contact with the ground. To see how fast the participants walked, the same distance for all activities was used.

Slow walk: The participants were instructed to walk slower than their normal velocity. Slow walking should be significantly slower than normal walk to make it easier to distinguish from normal walk.

4.2.2 Experiment execution

The experiment was performed at a corridor of the Norwegian Information Security Laboratory (NisLab) in A-building. The test-subjects wore the mobile phone attached to a belt. The mobile phone was placed on the right leg, by the hip for all test-subject. This is to ensure that mobile phone more or less has the same orientation. The distance for all activities, i.e., normal, fast and slow walk will be the same. The volunteers were asked to perform these activities several times for the same fixed distance of around 29 meters in only one session. One session includes walking fifteen different activities (normal, fast or slow). Each activity takes less than a minute to perform, a total form 7 to 10 minutes for each participants. The first sixth are fixed activities like *normal, fast, slow, normal, fast, slow*, for all subjects and the other nine are randomly chosen, an excerpt of these walks is shown in Figure 15.

The walking session was conducted in this way for all activities: For each session the test-subject was supposed to walk one of the activities, for instance normal walk, and then wait 3 seconds. After waiting 3 seconds the subject was asked to turn around and wait 3 more seconds before starting to do next activity (normal, fast or slow). The steps that a subject must follow are:

1. Wait 3 seconds.
2. Walk normal for the fixed distance.
3. Stop and wait for 3 seconds.

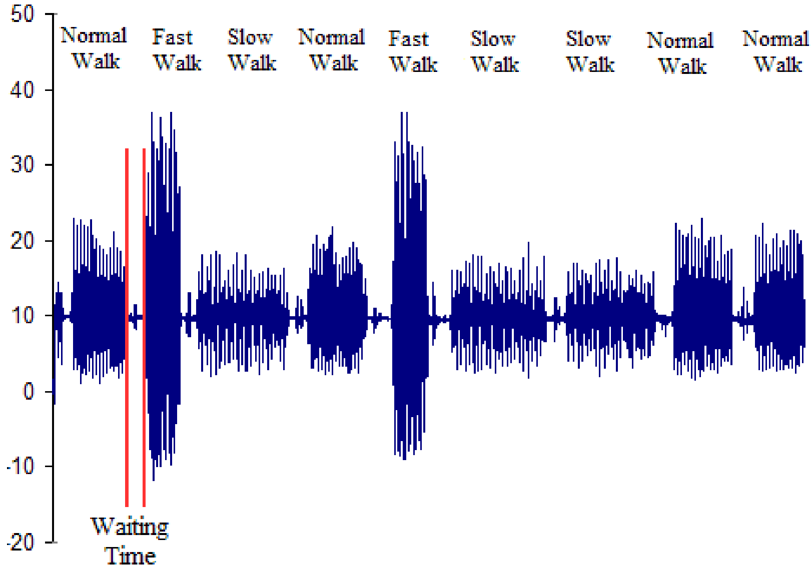

Figure 16: The signal of different walking activities.

4. Turn around.
5. Wait 3 seconds.
6. Walk fast (as it is shown in the Table) for distance back
7. Stop and wait for 3 seconds.
8. Repeat

The walking signal including waiting time is illustrated in Figure 16.

These steps must follow for each 15 activities, and after that we need to press the stop button to stop the data collection. Then the data are downloaded and store on the computer by using the USB port, also data are checked and verified.

Before the doing the experiment each participant had to sign an informed consent, see Appendix. We collected some information from each participant such as name, age and gender. In addition, we also write down for the kind of shoes that the participant has etc.

4.2.3 Volunteer crew

The volunteers in the experiment were students and employees at the University. In total, we had 45 participants (15 females and 30 males) who contributed in this experiment. They were all were healthy and with no injuries that may have effect in walking activities. All participants were of various age, height and weight and people from different countries and cultures. Most of them used shoes with flat sole, but we also had some with high-heels, slippers and winter shoes. The age range was from 9 to 59 years old. For males the age ranged was form 9 to 59, while for female the age ranged was from 19 to 38. Figure 17 illustrates the age distribution for all participants.

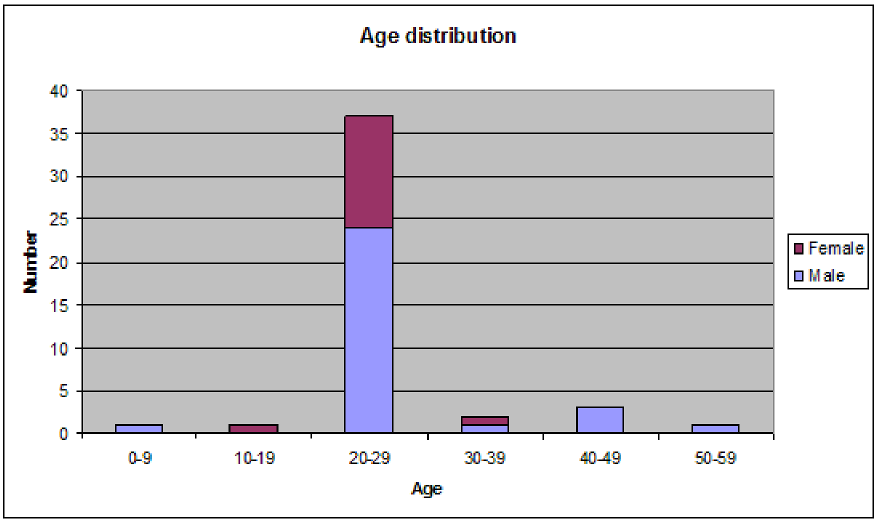

Figure 17: The gender and age distribution for the main experiment.

4.2.4 Environment

As mentioned earlier the experiment was performed in the corridor of NisLab in A-building. The experiment took place in the spring season and lasted for two weeks. The experiment was executed between 09:00 in the morning until 18:00 in the afternoon.

5 Processing and Analysis Details

A full automatic wearable sensor (WS) based gait recognition system using mobile devices is described in this section. The WS based recognition approach is the latest approach among the gait recognition methods available, i.e Machine Vision (MV) Based and Floor Sensor (FS) Based. WS is based on wearing motion recording sensors on the body of the person in different places; on the waist, pockets, shoes and so forth. Since wearable gait recognition system do not provide a full automatic mechanism today we will in this section we will give a possible solution to how this system is to be designed and implemented by including activity recognition as a major main step. Sensor based biometric gait research shows an increase in performance over time since 2005 where external dedicated sensors were applied until today where mobile phone accelerometers are being used. And to make gait recognition more stable, some issues need to be taken into consideration which we will see at the following subsections.

5.1 Scenario

Some examples on different scenarios where activity recognition and gait recognition would make phones more applicable as a security mechanism are provided here.

- **Shopping:** When a person is shopping, he or she is performs a lot of walking and standing. Since the user is constantly watching out for new equipments or clothing it simply means that the person is performing different activities by walking from one shop to another, from one cashier to another, etc.. In this case we can protect data of the person to ensure security of the phone.
- **Going to Work** PPeople go to work by different means of transport. Some people use their cars, bicycles or even their motorbikes. Since the user may have the mobile phone lost or stolen while walking out of the car or bicycling, it can ensure security. However, if a person is sitting in the car and the phone is standing still, the phone will also recognize that a "standing still" activity is ongoing, and thus the phone should not be used at all for authentication. In this case, a backup solution should be applied such as using the PIN-code.
- **Fitness/Jogging** Even when people are doing fitness, they might lose their phone when running outside their home. Running is still an activity and can also be used as a security mechanism towards authentication to the phone for usage.

These examples are only few out of many. An illustration of which activities can be recognized from gait signal data is shown in Figure 18. The interesting point here is that the mobile phone by using activity recognition for identifying activities and gait recognition for identifying the uniqueness of a person, together can establish a security link for mobile phone devices as an access control mechanism. To the best of our knowledge research has not implemented these two technologies into one full system. What

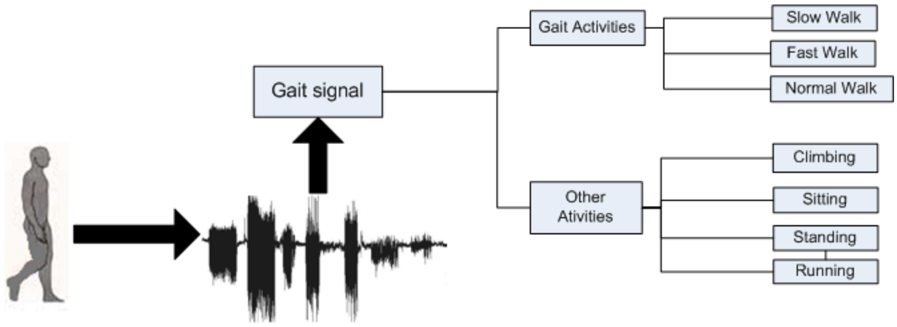

Figure 18: Walking and Non-Walking Activities.

we will see in the next subsection shows how we can apply activity and gait recognition approaches together and what is its performance like.

5.2 Design

The detection of everyday routine and leisure activities as we saw earlier like walking, running, sitting, and standing from gait signal recorded from wearable sensors make a step forward in the authentication. There have been studies done on gait recognition for authentication in mobile devices [110], but so far there are no studies in activity recognition for authentication. The data obtained from wearable sensors can be very useful for activity recognition as we have already seen in the recent section. Therefore, activity recognition is becoming a necessary step regarding continues authentication that is based on gait using wearable motion recording sensors in mobile devices. A proposal towards full gait recognition includes activity recognition. This simply means that a full automatic system includes:

- **Activity Recognition** Identifying activities from a gait signal where we only focus on stable activities, such as walking normal, slow or fast.
- **Gait Recognition** Extraction of the unique from the stable walking activities to be used for authentication on a mobile device.

Since a full gait signal consists of different activities, we propose to divide the activity recognition in two phases. First phase is a segmentation to find out where each activity's starting and ending point is located on the signal as illustrated in Figure 19. For this we propose the use of Sliding Windows, Top-Down, Bottom-Up and Sliding Window and Bottom-Up (SWAB) as referred to in section 3.2.3.

Second phase is the classification where we can see which activities are useful to forwarded to the gait recognition mechanism as illustrated in Figure 20. The classification task as can be seen in Figure 20 consists in itself that pre-processing before inputting the data for segmentation, is needed. After the segmentation process we apply feature extraction approaches. Feature extraction is the process of extracting the most relevant information form the data segments. The features extracted then passes through the classification stage. This stage includes the classification process of the data and creation of classifiers which are used to identify different human activities. For the classifications

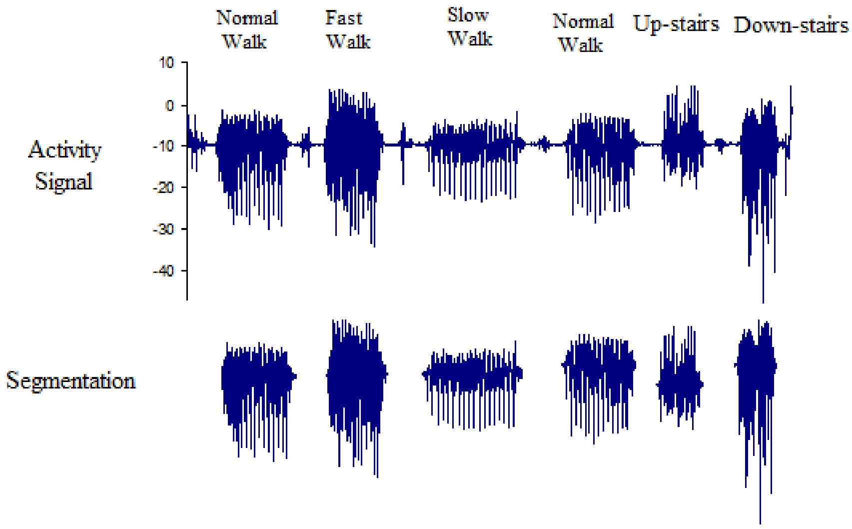

Figure 19: A full gait signal without segmentation (upper signal figure) and segmented walks (lower signal figure).

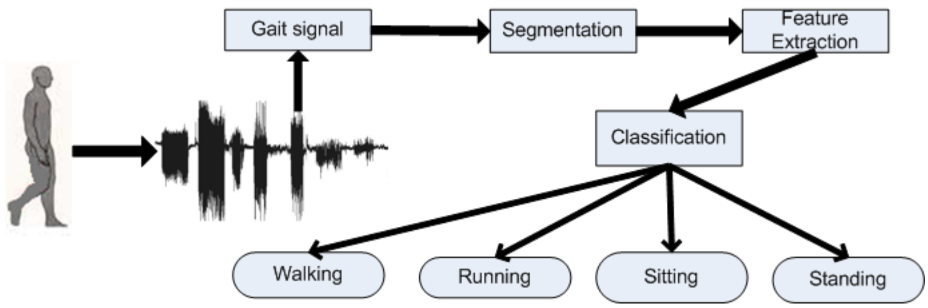

Figure 20: Classification of the Activities.

there are different approaches to apply. We thus propose to apply methods that are shown in section 3.2.3

Before explaining in details all the processes for both activity and gait recognition mentioned above, we will first show a novel system to be developed in order to understand why gait recognition system is strongly depended on activity identification for security reasons. Figure 21 illustrates a scenario on how the system should be used. Following the black arrow, we first perform the so-called template creation for different kind of activities. This is done in way that the subject is training the system, and thus, the different template creation for different activites, i.e., fast, normal, and slow walk templatesare stored into a database in a mobile devices. Next time the subject is going to walk with his mobile phone and is going to get authenticated (red arrow), the mobile devices initially extracts information about which activity has been performed. If the walk extracted was a normal walk, then we compare the normal walk probe template against the normal walk reference template in the database. In this case, we discard to compare the probe reference template with other than different types of walking templates. Thus, we ensure that the probability of false matches are lower than comparing the probe template against all templates in the database.

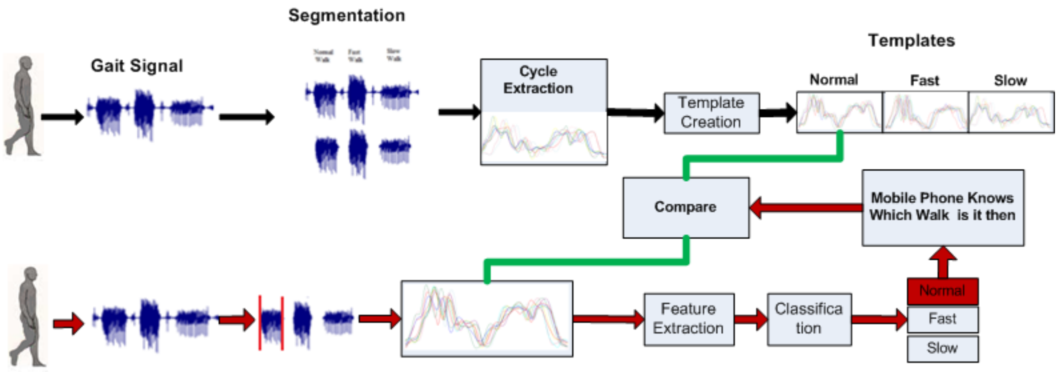

Figure 21: Authentication Process System. Black arrow indicates the process towards template protection. Red arrow indicates activity classification and green arrow the comparison.

5.3 Activity and Gait recognition

The process of gait and activity recognition as we explained above in this and previous section consists of several phases see Figure 20 and 21. In the following we will explain in more details which techniques and approaches are used in this project.

5.3.1 Segmentation

A visual description of the segmentation approach is illustrated in Figure 19. The data used are the resultant vector of all three (x,y,z) accelerations.

For the data analysis we have the gait signal consisting of 15 activities that need segmenting first. The experiment was set up in such a way that the relevant tasks were separated from each other by periods of inactivity, i.e. standing still, turning around and standing still again. In Figure 19 we can see this period of inactivity clearly between two activities as a more or less "flat line" with a small burst of activity in the middle due to the turning around.

The first step of the data segmentation for this data set has been performed by looking at activity over a short time interval. In particular for each datapoint we looked at the interval starting 25 samples before that data point to 25 samples after that data point (i.e. an interval of 51 samples representing 0.51 seconds of collected data). If the maximal difference in acceleration values collected in that interval was above a threshold, then we concluded that an activity was going on for that datapoint. This procedure was repeated for all datapoints that had at least 25 datapoints before and after it.

In the second step we looked at the intervals where we detected activity and inactivity and removed all intervals that were too short. For example when we found an interval of activity of length 1100, followed by an interval of inactivity of length 10 and then followed by an interval of activity of length 490, then we concluded that the interval of inactivity of length 10 was misclassified and the three intervals were combined to an interval of length $1600 = 1100 + 10 + 490$ of activity. In the same manner short intervals of activity between longer intervals of inactivity were removed. By removing these short intervals in increasing length we were also able to remove the short bursts of activity from the turning around in the middle of the "flat line" between the activities from the experiment.

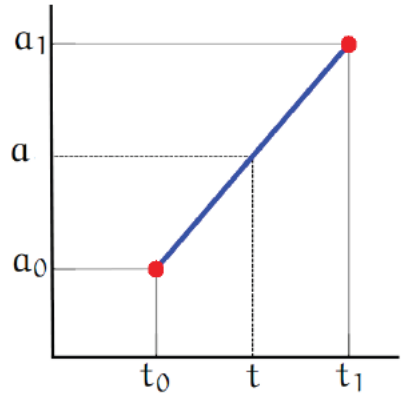

Figure 22: Given the two red points, the blue line is the linear interpolant between the points, and the value a at t may be found by linear interpolation [10].

5.3.2 Feature Extraction for Gait recogniton

There are roughly infinite ways to process and analyze the raw data that the sensor outputs. We will look at the main phases involved in most of the approaches and the algorithms we use in more details. For extracting the features for gait recognition we applied approaches and algorithms used in [10, 13, 24] with minor modifications. Before we extract features from the gait signal we must perform some pre-processing in the collected data in order to overcome some weaknesses with the sensor.

Time Interpolation: A limitation of the accelerometer sensor is that it does not precisely record one sample each 1/100 second. In order to obtain an observation or a value every 1/100 second we will apply some linear time interpolation in three axis data (x,y,z), because the time interval between two observation points are not always the same or equal. For instance, for two known observation points represented by the coordinates $(t_0; a_0)$ and $(t_1; a_1)$, the linear interpolant uses the straight line between these observation points. For any value of t in the interval $(t_0; t_1)$, the corresponding value a on the straight line between interval (a_0, a_1) can be found from Equation 5.1 . This is demonstrated in Figure 22. Therefore for all data values in the input we will apply this method. We start with the interpolation of data samples at t=0 in order to achieve the goal with a sample for every 1/100 second. In case a following sample occurs to have a value at the particular timestamp we do of course not change that sample. This technique of interpolating data is the simplest one and requires much less computational expense. More complicated alternatives for interpolations exist such as for example polynomial and spline interpolation. Polynomial interpolation is actually a generalization of linear interpolation, therefore we will have a linear function, but we can replace the interpolant to a polynomial of higher degree. As an alternative of applying a linear function for each of the intervals, one can apply low-degree polynomials in each of the intervals, and these polynomial pieces are chosen such that they fit smoothly together. Any such function that consists of polynomial pieces is called a spline, therefore the name spline interpolation. However the simplest method for linear interpolation of data is sufficient for our purpose.

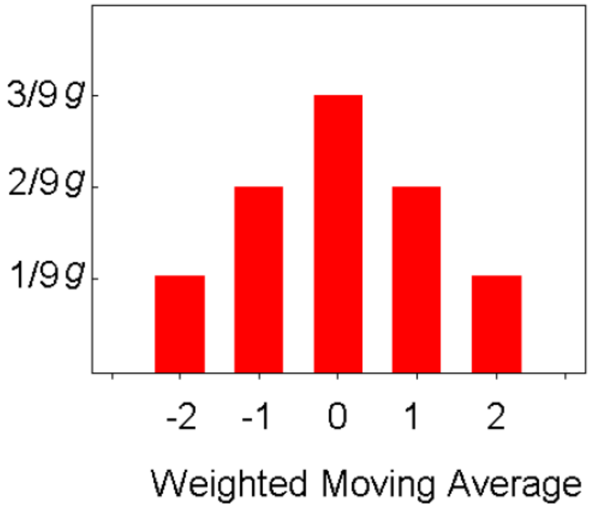

Figure 23: Weighted Moving Average [10].

$$\frac{a - a_0}{a_1 - a_0} = \frac{t - t_0}{t_1 - t_0} \Rightarrow a = a_0 + (t - t_0)(\frac{a_1 - a_0}{t_1 - t_0}) \tag{5.1}$$

Noise reduction: Another weakness of the data recorded form the accelerometer sensors is the noise. To remove the noise from the data there exist different methods. In this project we have used Weighted Moving Average (WMA). This method is fast and easy to implement. An illustration of this method is given in Figure 23. The only difference between WMA and other methods is in the weight of their neighbors. In WMA methods the nearest neighbors are more important than those more away, while in other methods all the neighbors have equal weight. The formula for WMA with a sliding window of size 5 is given in Equation 4.2. The processing is applied for are values in the input like with time interpolation excluding the first and the last two when we use a sliding window of size 5. As with time interpolation this is applied to all the values in the input we are processing excluding the first and the last two when we using a sliding window of size 5. Daubechies wavelet of order 8 was used in [22] to remove the noise, but in our project we have chosen to test only WMA with window size 5.

$$WMA\_a_t = \frac{(a_{t-2} * 1) + (a_{t-1} * 2) + (a_t * 3) + (a_{t+1} * 2) + (a_{t+2} * 1)}{9} \tag{5.2}$$

Where a_t is the acceleration-value in position t. The current values we are located at are given weight 3, the two closest neighbors' weight 2 and the next two neighbors' weight 1.

Cycle detection

An important phase is to detect the cycle starts and ends in the gait signal. There are different methods to split up a signal into periodic cycles such as splitting the signal into singular left and right steps [25, 35, 31] or take double steps [24]. In our project we have

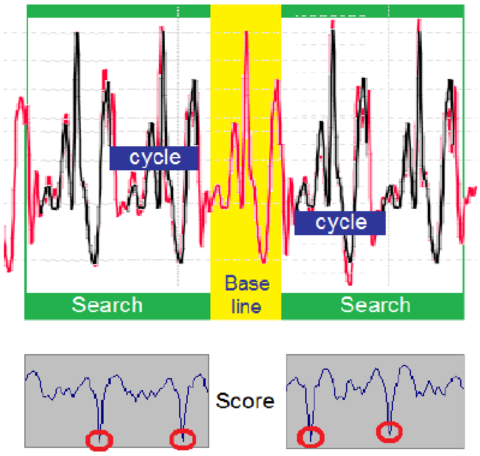

Figure 24: The yellow baseline area indicate the 70 samples that are extracted, the green area is the search area where the baseline is compared against a subset of the search area. The 4 black subgraphs are the baseline at those points it has the lowest distance with the search area subsets, and the difference between them (blue area) indicate the cyclelength[10].

chosen to take the double steps. A cycle contains double steps, and these are detected by looking at the minimum points in the cycle. The beginning of the next cycle will be at the end of the last cycle until to the final cycle of the gait signal. The cycle detection algorithm we have chosen has several sub-phases, therefore we will describe briefly each of the phases involved in the cycle detection. A short overview of what the step detection algorithm consists of is given in the following.

1. **Estimate cyclelength:**For more accurate detection of cycles, first we need to estimate the cycle length or how long a cycle is. After we estimated the cycles' length form the data collected we know that the length can range from 80 to 150 samples. The number of samples depends on the activity that the subject was performing. If we compare the fast walk with slow walk, we will see that the faster a person walks the less number of samples there are per cycle. It is the opposite with slower walk, the slower the person walks the more sample per cycle we have. The estimation of cycle length is performed by extracting a small subset of the collected data and then comparing this subset with other subsets that have the same length. The next step after the comparison of the subsets in a given search area the length of the cycle is estimated by using distance scores. This process is demonstrated in Figure 24.

2. **An indication of minimum value:**In order to find at what values the cycle detection should start, we need to get some information from the amplitude of the cycle. The information we need to get are minimum values. These values indicate expected values where the actual cycle detection should start.This process is demonstrated in Figure 25.

3. **Detect starting location:**After finding the minimum values, the last phase is to decide at which point we are going to start with the actual cycle detection. A normal starting point for cycle detection would be at the beginning of the data collected, but in our case for the reasons explained later in Section we will start at another point. We will start in the middle of the collected data. Due to this we have to find the most accurate

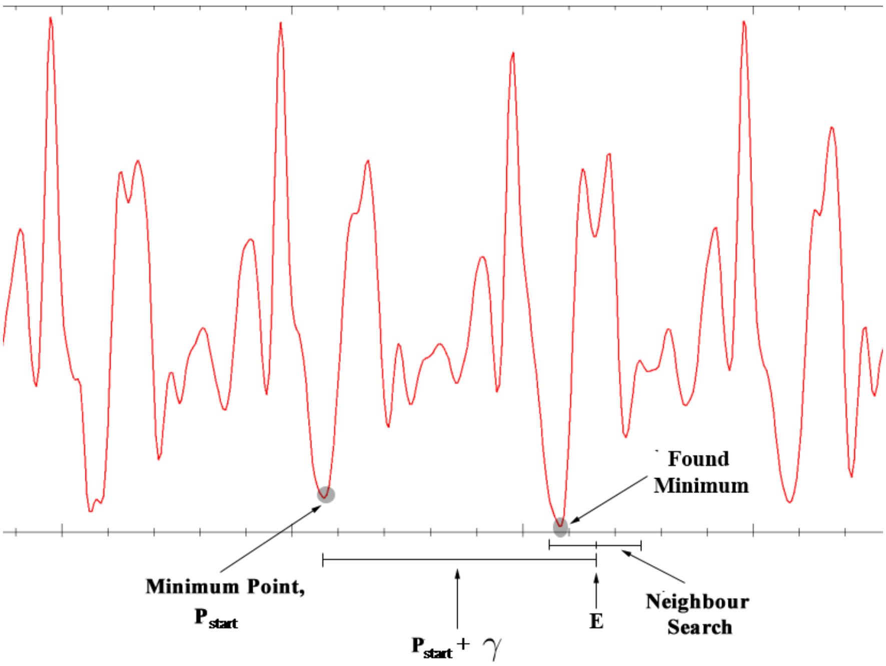

Figure 25: Cycle detection showing how each cycle (i.e the steps) in the resultant vector is automatically detected[13].

minimum point to start at.

4. **Detect the rest of the steps:** Finally after going through previous phases and finding the minimum points we are ready to start with the actual detection and able to find the beginning and end of each cycle. This is done by first searching cycles forward from the starting location point detected in the previous phase, and when forward searching is complete we repeat this process by searching backwards.

The cycles extracted for normal, fast and slow walk from the gait signal are illustrated in Figures 26,27 and 28.

5.3.3 Feature Extraction for Activity recognition

For activity recognition we need to select and calculate individual features for each activity. Feature extraction for activity recognition is a very important step. They need to be carefully chosen due to strong influence in the result of final classification. In previous section we described the feature extraction for gait recognition which was done by extracting cycles form the gait signal. For activity classification we extracted features from the extracted cycles. Here we extracted four and eight features form the cycles. In Appendix C, a full implementation of the calculations of the features are shown for more details. The features extracted were:

1. **Standard Deviation:** Let X $=x_1, x_2, x_3 \ldots x_n$ be a random variable with mean value μ: $E[X]=\mu$
   Where the operator E represent the average or expected value of X. Then the standard deviation of X is:

$$\sigma = \sqrt{E[(X-\mu)^2]} \tag{5.3}$$

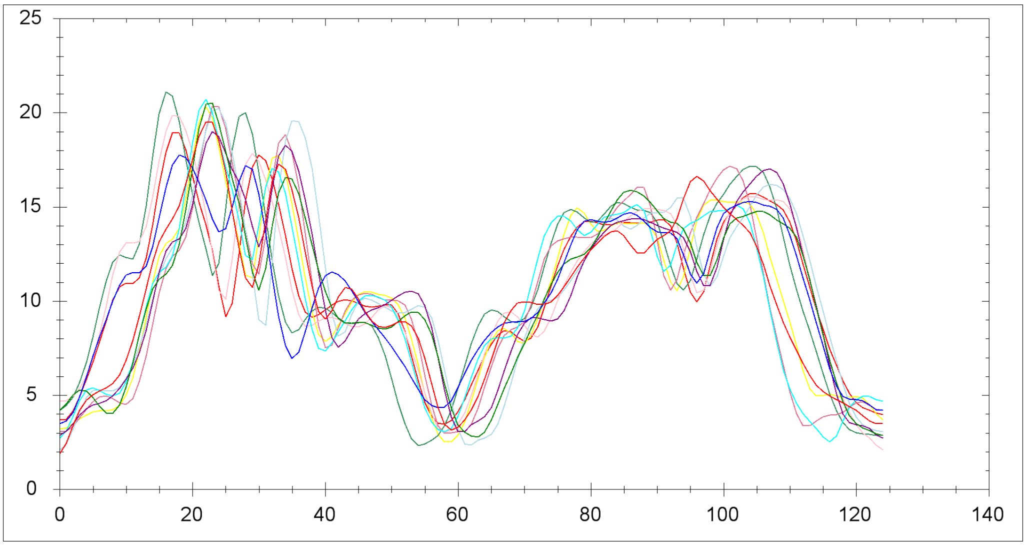

Figure 26: The cycles extracted from normal walk.

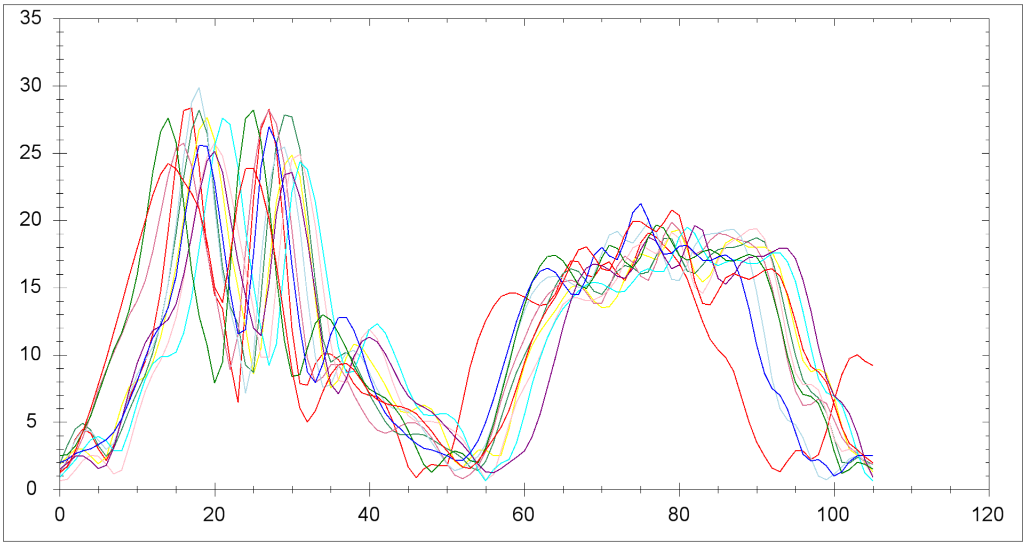

Figure 27: The cycles extracted from fast walk.

2. **Minimum Value:** is the minimum value of X $=x_1, x_2, x_3 ... x_n$:

$$MIN_X = min(X) \tag{5.4}$$

3. **Maximum Value:** is the maximum value of X $=x_1, x_2, x_3 ... x_n$:

$$MAX_X = max(X) \tag{5.5}$$

4. **Cycle length**: is the length of the cycle extracted.

5. **Root Mean Square:** In the case of a set of n values $x_1, x_2, x_3 ... x_n$ root mean square

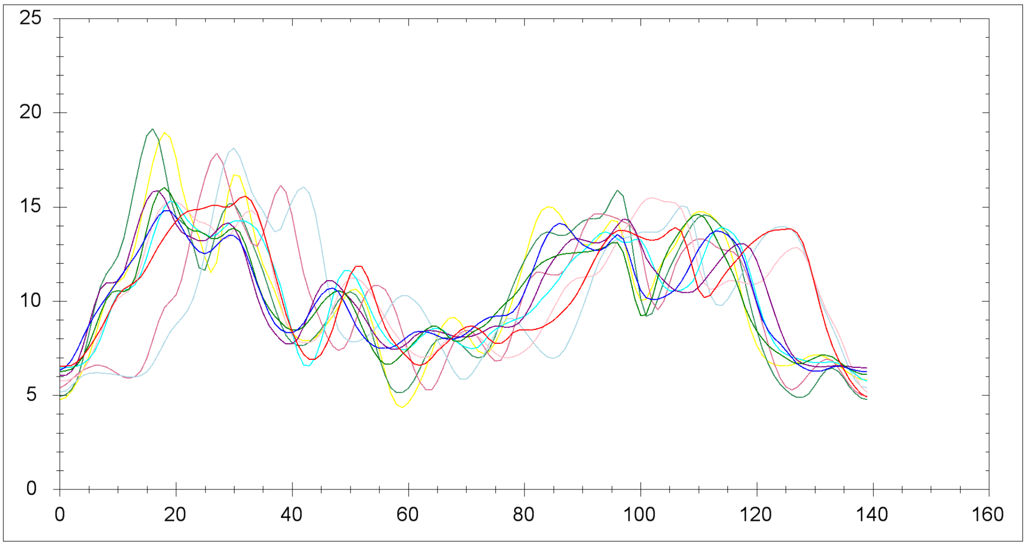

Figure 28: The cycles extracted from slow walk.

is defined as:

$$RMS = \sqrt{\frac{x_1 + x_2 + x_3 + ... + x_n}{n}} \tag{5.6}$$

6. **Mean:** the mean value of of X $=x_1, x_2, x_3...x_n$ is:

$$\mu = \frac{1}{N}\sum_{i=1}^{n} X_i \tag{5.7}$$

7. **Entropy:** The entropy was computed with equation:

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i). \tag{5.8}$$

8. **Energy**

$$H(x) = \sum_{i=1}^{n} (x_i)^2. \tag{5.9}$$

The reason why we chose these features is because each of them outputs different values for different activities and variates from user to user. As it is shown in Figure 29, acceleration values differ for different activities, and standard deviation is one of the features that was used to capture this fact. Min and max capture the minimum and maximum value of a cycle. The last feature calculated is cycle length. This is done due to different cycle length for fast, slow and normal walking activities. First we will test the machine learning algorithms with four features and then we will test them with eight features to see if it improves the classification results.

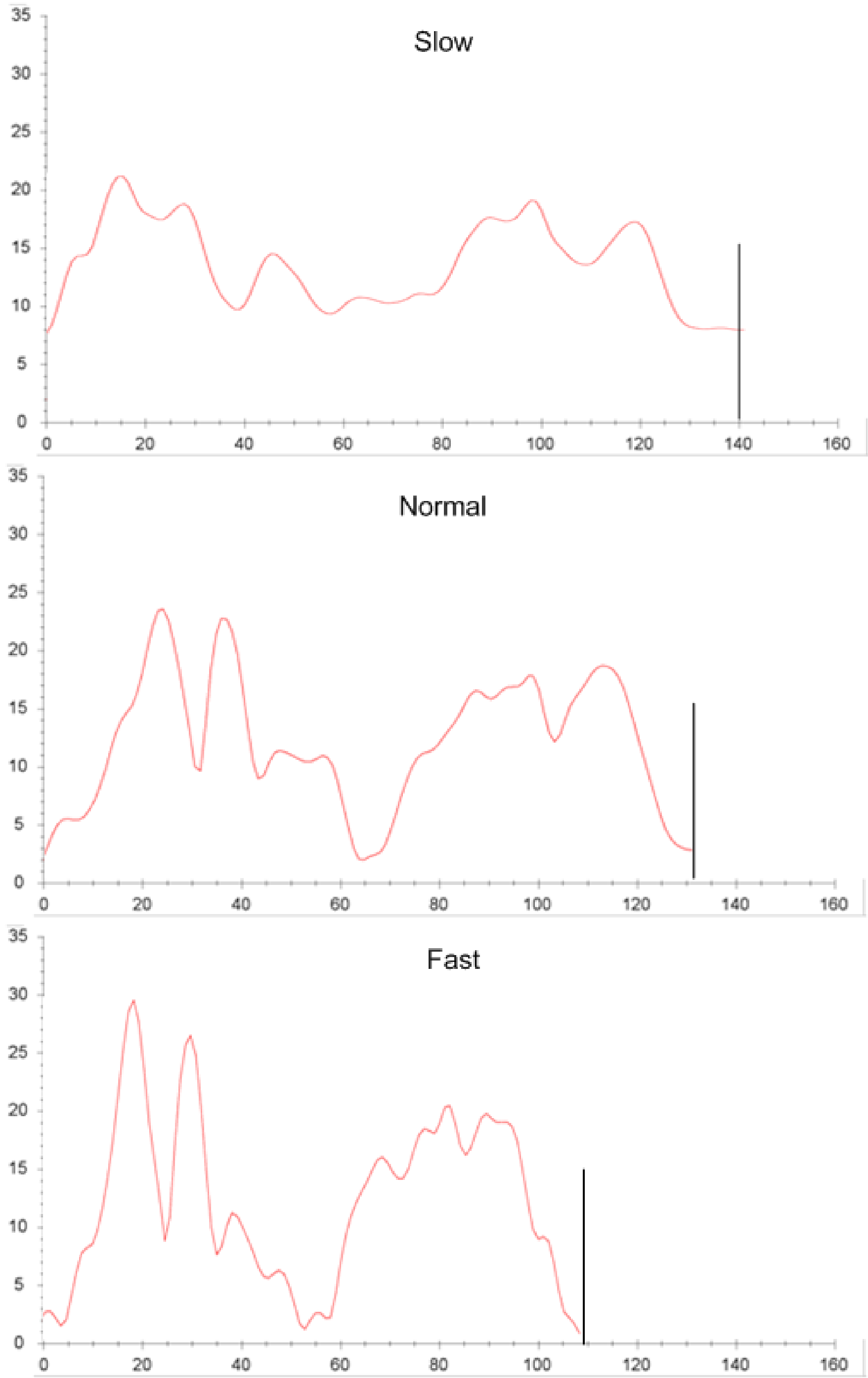

Figure 29: Average cycles for slow, normal and fast to see the to see the difference between them.

6 Analysis and results

In this chapter we will describe the data analysis process and we will report on the results obtained from the gait recognition and activity classification. This chapter is divided into two sections. First section explains the how the data has been analyzed for activity classification and the results of classification accuracy. Second section describes how the data has been analyzed for gait recognition and the results of gait performance.

6.1 Activity Recognition Analysis

The activity recognition analysis is based on the data from the feature extraction as explained in previous chapter. This task is carried out by using an open source software by name WEKA which programming language is Java and developed at the University of Waikato. WEKA has implemented a collection of machine learning algorithms which contains tools for data pre-processing, classification, regression, clustering, association rules etc.. The input file format for WEKA is an .ARFF format file. Thus, in order to use WEKAs functionalities for the classification task, a tool was first written C# class to output the features as attributes in an .arff file (weka). Some examples of this file are attached in the Appendix. The Arff file contains a header and a data section . The header for the arff file contains the name of relation and a list of attributes (extracted features) and their data type, as shown in Figure 30.

```
@relation features

@attribute stddev real
@attribute min real
@attribute max real
@attribute clength real


@attribute class {Norm,Fast,Slow}

@data

7.05862070139076    1.2563081354139 25.3789723665564    106 Fast
7.93199296329994    0.97077874551533    30.0940660316425    106 Fast
7.2930599669421 1.18272773357428    28.9903719401684    106 Fast
7.12937822482773    0.880178430614616    26.8366093184836    106 Fast
7.1669598874395 0.588775719683462    29.4569532401196    106 Fast
7.72334544757136    0.573402849391255    28.4354720755501    106 Fast
8.40712999338158    0.663540537309079    33.595337017504 106 Fast
7.40975905705479    0.781139005477846    29.2958840095302    106 Fast
7.80779867854086    1.42353397168901    30.3696876079135    106 Fast
7.36013831796895    1.30988040721656    30.0369298199602    106 Fast
8.79661635925904    0.822107399123648    35.9936122833634    106 Fast
5.41948898859138    2.09699451905214    23.9570663157556    125 Norm
4.92328174820974    4.11534146115936    22.1108117825232    125 Norm
```

Figure 30: Arff file content.

To make it easier for users to use WEKA it provides both a rich graphical user interface and a powerful command-line interface. Working with this tool is straight-forward, mainly due to its GUI Explorer as shown in Figure 31.

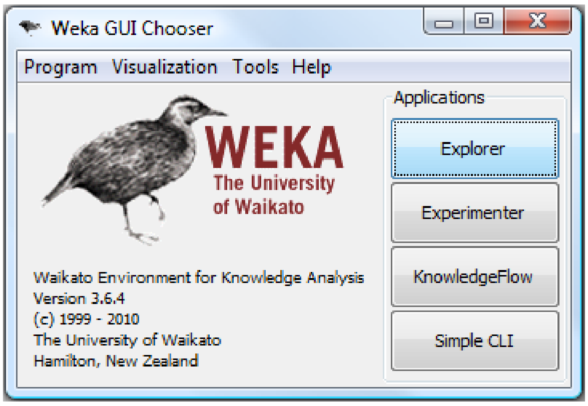

Figure 31: Weka GUI.

WEKA's GUI has four main options which appears at the beginning of the execution.

- **Explorer :** An environment for exploring data with WEKA such as preprocessing, attribute selection, learning, visualization etc.
- **Experimenter:** This environment is used for performing experiments such as testing and evaluating machine learning algorithms.
- **KnowledgeFlow:** Provides the same functions as the Explore environment but with a drag-and-drop interface.
- **SimpleCLI:** This environment provides a simple command-line interface for typing Weka commands for operating systems [111].

For our analysis we chose the "Explorer environment" since it performs all of the classification tasks needed. There are three different types of performance evaluation testing, that the user now can choose, see Figure 32:

- **Cross-validation (default):** performs n-fold stratified cross-validation, when the number of folds is by default 10.
- **Train/Test Percentage Split (data randomized):** It splits the dataset based on a given percentage for training and testing.
- **Train/Test Percentage Split (order preserved):** the user priory splits the dataset in two files train/test and uses these files for training and testing [111].

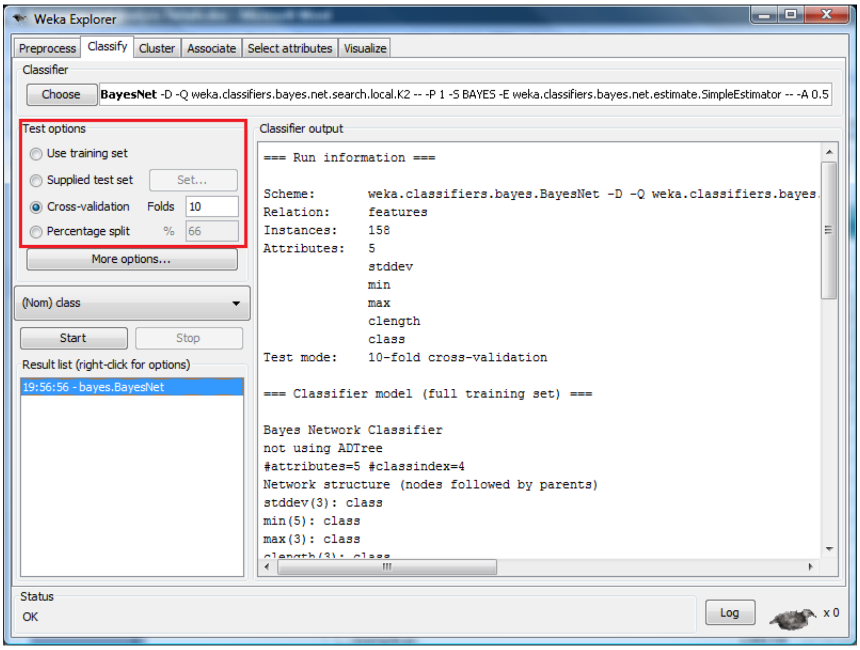

Figure 32: Experiment types.

6.1.1 Results

With the extracted features from fifteen session per subject where each session consisted of one of three different walking activities (normal, fast and slow) we did do different types of testing for the performance evaluation. Currently, we will show which combinations of the data we performed, how data are split for training and testing and why we did it that way.

The performance evaluation combinations:

1. **Cross validation:** k-fold cross-validation uses k-1 folds for training and the remaining for training. Cross validation by using WEKA splits the train/test data by choosing randomly. An example of one iteration with a 3-fold cross validation is as shown in Figure 33. The first performance evaluation we did was cross validation for individual-based activity recognition. This means that we look separately at each user's activity performance. Table 19 shows the results of classification by using different classifier.

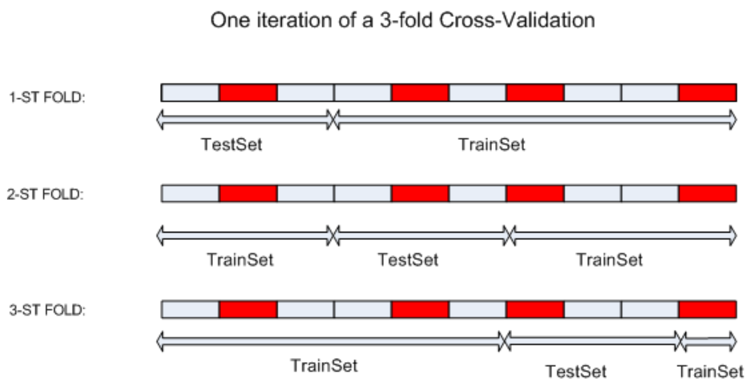

Figure 33: 3-fold cross-validation with 10 samples.

Table 19: Personal Cross Validation

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	94.88%	94.53%
NaiveBayes	89.31%	88.14%
LibSVM	92.59%	N/A
MultilayerPercepton	92.77%	93.70%
RBFNetwork	91.98%	90.34%
RandomTree	93.87%	93.84%
LMT	96.08%	96.23%

Table 19 shows the classification accuracies using different approaches based with either 4 and 8 features selected. In most cases we achieve high classification using cross validation. In other words, it means that we can easily distinguish all activities from each other, such that a application system implemented can output which walk has been performed at a certain time. From all of the machine learning algorithms evaluated for with the four features, the best result was given by LMT (Logistic Model Trees) with an accuracy of 96.08%, also an accuracy of 94.88% was achieved by BayesNet. It is obvious that the accuracy rate of 96.08% and 94.88% indicate how useful these two algorithms are for correctly identifying different activities. Another test with 8 features was also performed and evaluated. We wanted to investigate the change of performance and how the amount of different features would affect the performance. As can be seen in Table 19, we do not see an significant change in performance. Sometimes the performance with 4 features gives better results. This is due to the fact that the selected features are not so suffiecient for the evaluation.
WEKA also generates a confusion matrix which shows the number of activities identified correctly and incorrectly. For example, Figure 34 shows that fast and slow walks were recognized 100% correctly. An excerpt of the results from the cross validation accuracy are shown in Figure 35 and the full table is shown in Appendix. The average performance is more than 90% with most rates which can be categorized as an acceptable range for application use.

	BayesNet	NaiveBayes	LibSVM	MultilayerPercepton	RBFNetwork	RandomTree	LMT
Nr							
1	96.8354	98.1013	96.2025	98.1013	98.1013	96.2025	98.1013
2	92.0863	89.2086	85.6115	89.9281	91.3669	92.0863	92.8058
3	98.0198	96.5347	95.0495	97.0297	97.0297	98.5149	98.5149
4	95.1515	93.3333	96.3636	95.7576	96.9697	98.7879	98.7879
5	97.973	91.8919	95.2703	96.6216	93.9189	95.2703	98.6486

Figure 35: Excerpt the results from the cross validation accuracy.

2. **Personal testing/training :** The second test is labeled as personal testing/training and illustrated in Figure 36. Manually we have split the individual-based data into training and testing. We did three different tests for personal-based combination. The first was named 3+12 which means that three first sessions were used for training the classifiers and the remaining sessions for testing, in this case 12 remaining sessions. The second was named 6+9 where 6 first sessions were used for training and the 9 remaining sessions for testing. The last was named 9+6, where as training data set we used first 9 sessions and the rest of the data for testing. Results of these test are shown on Tables 20 21 22. For these three test we evaluated only three classifiers since we

```
Time taken to build model: 0.49 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         155               98.1013 %
Incorrectly Classified Instances         3                1.8987 %
Kappa statistic                          0.9713
Mean absolute error                      0.0335
Root mean squared error                  0.1143
Relative absolute error                  7.5986 %
Root relative squared error             24.3687 %
Total Number of Instances              158

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.952     0         1         0.952     0.976      0.992    Norm
                 1         0         1         1         1          1        Fast
                 1         0.028     0.942     1         0.97       0.994    Slow
Weighted Avg.    0.981     0.009     0.982     0.981     0.981      0.995

=== Confusion Matrix ===

  a  b  c   <-- classified as
 60  0  3 |  a = Norm
  0 46  0 |  b = Fast
  0  0 49 |  c = Slow
```

Figure 34: Confusion matrix generated by weka.

saw that these were performing best from the previous results. Results on the Tables 20 21 22 show different accuracies for each of the test performed. An interesting fact noticed is that when we compare 3+12 with 6+9 , we see an significant increase from 65.52% to 78.97% by using MultilayerPercepton classifier. From these two results we can conclude that recognition of the activities depends on the amount data used for training. The more data the better the classification accuracy is. Another interesting fact is on the results of 6+9 and 9+6 tests where there is a slight increase in accuracy of 9+6. Therefore, it is important to find the balance of how many training data is needed.

	Training	Testing
1	3 Sessions	12 Sessions
2	3 Sessions	12 Sessions
⋮	⋮	⋮
45	3 Sessions	12 Sessions

Figure 36: Example of Personal Traning/Testing test.

Table 20: Personal testing/training 3-12

Classifier	**BayesNet**	**MultilayerPercepton**	**LMT**
Accuracy	68.83%	65.52%	64.54%

3. **Global Cross Validation:** Third combination performed was labeled as the global cross validation. In this test we merged all data together from all sessions of all 45

Table 21: Personal testing/training 6-9

Classifier	BayesNet	MultilayerPercepton	LMT
Accuracy	77.36%	78.97%	75.26%

Table 22: Personal testing/training 9-6

Classifier	BayesNet	MultilayerPercepton	LMT
Accuracy	78.35%	81.03%	78.62%

subjects into one file. The results are shown in Table 23. In contrast to the personal cross validation, these results shows how similar or different each subjects fast,slow, and normal walk is from each other for all users. From a performance point of view, we would like to strive after higher accuracies. The results shows that the LMT and LibSVM performs better with an average recognition rate of 79.62% and 79.58%, for four features. This clearly shows that the recognition accuracy is lower compared to cross validation used for individual-based activities classification. This is due to the fact that some peoples normal walk might look like other peoples slow or fast walk, vice versa. Still, we are very satisfied with these high results and we expected ofcourse lower results, since peoples walking types and speed are very dissimilar so that we would have overlaps. For 8 features we observe again that the results are lightly increasingly changed.

Table 23: Global Cross Validation

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	73.62%	72.94%
NaiveBayes	71.57%	72.29%
LibSVM	79.58%	N/A
MultilayerPercepton	73.13%	75.41%
RBFNetwork	72.87%	72.71%
RandomTree	75.16%	76.33%
LMT	79.62%	80.21%

4. **Session-individual:** In this combination we performed two sub-tests named 3+12 and 6+9. For the 3+12, all data from the three first sessions were merged in a file which was used for training as it is illustrated in Figure 37. Then for each subject we created a file with 12 remaining session data for testing. For the second test 6+9 we used all the data from first six sessions for training and 9 remaining sessions for each subject we test the classifiers. he results are shown in Tables 24 25. The results are almost the same for both tests.

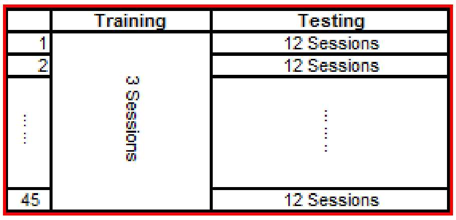

Figure 37: Example of Session-individual test.

Table 24: Session-Individual 3-12

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	68.10%	69.05%
MultilayerPercepton	69.44%	71.48%
LMT	64.96%	67.03%

Table 25: Session-Individual 6-9

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	68.38%	68.25%
MultilayerPercepton	68.43%	70.92
LMT	66.63%	67.85%

5. **Session-Based:** This test is based on sessions and illustrated in Figure 38. We performed three different sub-tests in this combination. We called these test Session based 3+12, 6+9 and 9+6. In the 3+12 we take the three first sessions from all users and merging them into one file to be used for training. The 12 remaining session are merged into another file and used for testing. We did the same for 6+9 and 9+6 where the first number indicates the number of sessions used for training (6 sessions for training + 9 sessions for testing and 9 sessions for training + 6 sessions for testing) and the second for testing. The results from these three tests are shown on Tables 26 27 28. These accuracies show the differences of all users normal, fast and slow walks. As we can see from the results there are some slight increases in the accuracies. For some classifiers we observe that the more sessions used for training the better the results of classification are. However, for one of the classifiers *MultilayerPercepton* we see the opposite, even though it performs better for test 3+12 and 6+9. This is probably due to the fact the the training of the system probably had founds its best threshold with the first mentioned.

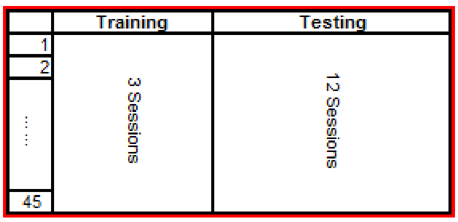

Figure 38: Example of Session-Based test.

Table 26: Session Based 3-12

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	68.28%	68.76%
MultilayerPercepton	69.19%	71.33
LMT	64.97%	66.75%

Table 27: Session Based 6-9

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	68.55%	68.48%
MultilayerPercepton	68.77%	71.24
LMT	67.16%	67.80%

Table 28: Session Based 9-6

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	68.11%	68.86%
MultilayerPercepton	67.37%	70.79%
LMT	67.62%	69.67%

6. **User-Based:** All previous tests except the global cross-validation were to test the performance of the classifiers based on users sessions. The user-based test is a user dependent test and illustrated in Figure 39. Based on the users we split the test into 5+40, 10+35 15+30 and 20+25. The number 5+40 means that the data from the first five are used for training and the 40 remaining used for testing. For the test 10+35, the data of the first ten users are merged iton one file for testing and the remaining data merged into one file for testing. Tables 29 30 31 32 show the result of these tests. These results indicate that we have a slight increase in the results when the number of users for training is higher. These accuracies show the differences of all users normal, fast and/or slow walk. It is a general description so that we can have an estimation of how much alike or dissimilar peoples (normal, fast and slow) activities are from each other.

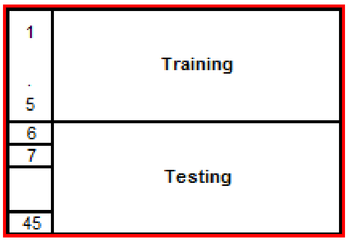

Figure 39: Example of User-Based test.

7. **User-Based Individual:** Last test performed is based on user where a number of amount of users are used for training and the remaining users each are separately tested against the training model, see Figure 40. Three different tests were performed for this combination: 5+40, 10+35 and 20+5, which means that the first 5 users were merged into one file and we used for training and 40 remaining users

Table 29: User based 5+40

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	63.5%	67.78%
MultilayerPercepton	65.6%	57.86%
LMT	61.13%	57.98%

Table 30: User based 10+35

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	64.03%	65.44%
MultilayerPercepton	67.31%	62.46%
LMT	64.03%	58.94%

Table 31: User based 15+30

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	67.59%	68.16%
MultilayerPercepton	62.9%	63.56%
LMT	63.51%	62.74%

Table 32: User based 20+25

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	69.99%	68.16%
MultilayerPercepton	70.45%	72.71%
LMT	64.51%	59.84%

individual were used for testing separately form each other and see if the activities can be classified based on the training. These results are shown in Table 33 34 35. We can see from the results that when the number of users for training is higher then we have a slight increase in the accuracies. In conclusion we can say that the more data we use for training the better results we have.

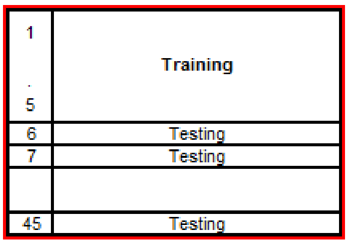

Figure 40: Example of User-Based Individual test.

Table 33: User-Based Individual 5-40

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	63.29%	67.46%
MultilayerPercepton	66.08%	58.23%
LMT	61.15%	58.39%

Table 34: User-Based Individual 10-35

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	63.64%	65.18%
MultilayerPercepton	66.95%	62.52%
LMT	63.67%	58.99%

Table 35: User-Based Individual 20-25

Classifier	Accuracy with 4 Features	Accuracy with 8 Features
BayesNet	69.51%	67.63%
MultilayerPercepton	70.08%	72.48%
LMT	64.10%	59.79

6.2 Gait Recognition Analysis

In this section we will explain the process of data analysis for gait recognition. Gait recognition performance evolution consists of also two phases. First phase is the extraction of gait cycles from each walk that was segmented. Second phase explains how we compare the features against each other. In this case we are not applying machine learning approaches, but instead we use distance metrics. Then we will explain how we compute distance scores. First phase is described in more details in pervious section and the second phase we will explain in more details below. Each user performed 15 walks. From each of the 15 walks the cycles are extracted as described in Section. For each walk we create the feature vector consisting of all of the extracted cycles stored in a template.

6.2.1 Feature Comparison

Since each feature vector template has dissimilar lengths we were unable to use distance metrics such as the Manhattan or Euclidean. Instead we applied a time series analysis named the Dynamic Time Warping (DTW) which is an algorithm for measuring similarity between two sequences which may vary in time or speed.

A new modified distance metric, named the Cross-DTW metric (CDM), is applied. This metric cross-compare two sets of cycles to find the best matching pair for vectors or unequal length:

Cross Comparison: is used to find the most optimal and best distance score when cross-comparing two set of cycles, denoted $C^S = \{C^S_1, ..., C^S_N\}$ and $C^T = \{C^T_1, ..., C^T_M\}$. This simply mean that each cycle in set C^S is compared to every cycle in set C^T. The comparison distances are calculated by the Cross-DTW metric (CDM). From the total number of N $\times$ M similarity distance scores gained, the minimum distance score is selected,

$$d_{min} = min\{CDM(C^S_i, C^T_j)\}$$

where i=1..N and j=1..M. The pair of cycles with the most minimum similarity score is considered the best matching pair. Thus, this best (i.e. minimum) similarity score, d_{min}, is used as the similarity score between set C^S and C^T.

The output of the CDM is called the distance score S, where a low value of S indicates high similarity, while a high value indicates low similarity.

6.2.2 Creating distance score table

As mentioned in earlier chapters, each participant performed 15 walks which include normal, fast and slow, for each walking activity we create a template, which means three templates for each participant duo to three different walking activities. The templates contain the cycles extracted from the walking signals. Next step after creating the templates for each 45 participants and choosing the distance is comparing all normal templates against each other , all fast templates against each other and the same for slow as well. An example of this comparison is when the normal template of one person is compared against the same person's normal walk which is known as a genuine attempt and also against the all normal walks of the other participants which is known as an imposter attempts. Table 36 shows a small excerpt of a comparison score table.

	P_1S_1	P_1S_2	...	P_1S_K	P_2S_1	P_2S_2	...	P_NS_K
P_1S_1	-	-	...	-	-	-	...	-
P_1S_2	G	-	...	-	-	-	...	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
P_1S_K	G	G	...	-	-	-	...	-
P_2S_1	I	I	...	I	-	-	...	-
P_2S_2	I	I	...	G	G	-	...	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
P_NS_K	I	I	...	I	I	I	...	-

Table 36: Comparison scores. G = [genuine], I = [impostor], P = [subject-ID] and S = [session-ID].

The number of genuine attempts and impostor attempts we can calculate by using the Equation (1) and (2):

$$G_(tot) = \frac{S * (S - 1) * N}{2} \tag{6.1}$$

$$I_(tot) = \frac{S^2(N - 1) * N}{2} \tag{6.2}$$

- **Genuine recognition attempts:** The template of each user is compared to the remaining template of the same user, but avoiding symmetric matches (i.e. if the template j is matched against template k, template k is not matched against template j);
- **Impostor recognition attempts:** The template of one user is compared to the remaining templates, but different users and avoiding symmetric matches.

After scores have been calculated, we can now initiate the creation of Decision Error Trade-of (DET) curve for each score set. The DET curve as we explained earlier shows the trade- of between the False Match Rate (FMR) and False Non-Match Rate.

6.2.3 Results

Three different performance evaluations were calculated. We calculated the equal error rates (EER) of all three and compared normal against normal, slow against slow, and fast against fast. The comparison approach is described in subsection 6.2.1 and the results retrieved are shown in Table 37

Table 37: Performance Evaluation (EER) of Gait Recognition

Fast	**Normal**	**Slow**
15.48%	17.17%	21.07%

What we observe from these three results is that by walking fast we retrieve better performance with a difference of almost to the worst 6%. One reason for this retrievement of the EER is the wearing of different shoes on the different sessions. Further influencing factors are the worn trousers which have an impact on the position (height, angle etc.) of the bag as well as on how stable it was attached.

7 Discussion and Future Work

In todays highly use and development of mobile devices (mobile phones, PDA, tablet computers, etc.) and all the opportunities they offer, makes the privacy and security concern issue raise. Latest generation of mobile phones provide excellent opportunities such as huge storage capacity and processing capability. Due to that they are not used only for storing telephone numbers, sending text message or used for mere communication. These devices are today used for storing more sensitive data such as e-mail, addresses, appointment, photos, financial document and other important document that contains confidential information. As a result of this, keeping all information secure is becoming highly important. If an device has been lost or stolen, all information becomes accessible for theft or strangers. Furthermore, the mobile phone can be used for making calls without limits until the owner informs the provider. In order to decrease the risk to the owners security and privacy, mobile phones should continuously authenticate the holder of the device. Current methods used for authentication are based on PIN codes, and sometimes the PIN-codes are not used at all. A weakness of using PIN codes and passwords is that people usually use these in different application for authentication. Due to this they need to remember loads of passwords. As a consequence of this problem for people is forgetting. Another problem is that people choose the easiest way by authenticating themselves only at the switch on time. A survey showed that only 66% (no PIN-authentication required after stand-by phase) of the respondents use PIN-authentication only at switches on and only 18% of the users also use the standby mode authentication [110]. Using PIN codes for standby mode authentication is too much effort, every time when the users need to use the mobile phone he/she had to press the PIN, therefore people usually chose the easy way. Also passwords are something that can be lost and forward to someone else.

Since mobile devices at present time do not offer continues user authentication, a solution for authentication in mobile phones could be using gait and activity recognition. People usually do different activities during the day, for example when they are shopping, they perform a lot of walking activities, therefore activity recognition is becoming a necessary step towards continues authentication based on gait using wearable sensors in mobile phones.

Current gait recognition research has had its focuses on manual extraction of features. A solution to this issue could be activity recognition that would reduce the disadvantages of gait recognition by identifying daily activities and authenticate the user continuously and automatically. Since a full gait signal consist of different activities, activity recognition would make not it possible to identify walking activities such as normal, slow and walk and use these for authentication but also it will identify passive activities such as sitting, standing still and will avoid authentication. Even tough that we have seen that activity recognition is an important step during this project there are still open questions and problems to be dealt with in the future.

Gait and activity biometrics are behavioral biometrics, and they offer continues authentication where the user will not notice it and also higher security of data in mobile devices. In a real application scenario every user first has to create templates for different activities like normal, fast and slow walk and the authentication process will be based on these. When the user is performing these activities, the mobile phone will automatically compare these walkings with the stored templates, and if they match the user can have access in the device otherwise it will denied the access. For passive activities as a backup method can be used PIN codes.

This project we have been dealing with is quite new area to the best of our knowledge with mobile devices and for the first time ever, we have been able to show how activity recognition can be used for gait recognition. As a new area there is still a lot of aspect that would need further research. Activity recognition is also a new area of study and as we saw in the related work, in last decade is becoming an interesting research field duo to its application in many areas. Many researchers have tried to recognized different daily activities in their research. In our experiment we included only stable walking activities like normal, fast and slow, it would be more natural to include more circumstances like, walking upstairs and downstairs, walking up- or downhill. Another interesting research topic would be looking at different environments and performing an experiment under normal circumstances and not only controlled laboratory settings where subject are told how to walk. Another challenge and interesting aspect for data collected under normal circumstances would be the segmentation process, for instance the user is walking normal and suddenly he or she starts walking fast. Implementing different segmentation methods to see which is performing the best to find boundaries between these activities would be interesting. In addition to improving the classification accuracy, an interesting research would be to test various classifiers with different set of features and to see the effect of increasing the number of features in the overall accuracy.

8 Conclusion

Mobile devices are often lost or stolen in a state in which they can be used without any authentication. Regular user authentication on mobile devices is required because they are currently not well protected in a working state by having only an explicit authentication, and users are not willing to perform explicit authentication very often. The method for the protection of mobile devices proposed in this thesis is based on activities from the gait patterns. The use of gait as the biometric modality in mobile device is very natural because users frequently walk carrying their mobile in all different environments. And if something suspicious happens during the walk, the gait recognition modality will be activated, by first performing activity identification to ensure improved performance of security and privacy. An embedded accelerometer allows mobile devices to gait recognition frequently and unobtrusively, as a background task, while the user is walking and carrying the mobile device. This would protect mobile devices in "normal" state, but would be vulnerable to different walking environments, especially if a person is walking unnaturally. Since there is a fairly high risk of mobile devices being lost or stolen in urban environments such as streets, public transport or shopping areas, the implementation of of activity identification with unobtrusive biometrics, gait, should be beneficial for protecting user privacy

In this research project we have looked at interesting aspects of the biometric feature gait and its application to be developed by using activity recognition. A novel, simple and continuous rich automatic authentication system on mobile devices has been proposed and some parts analyzed, i.e. the classification. Activity and Gait Recognition has been studied separately in the recent years, but the interest has become so high lately when mobile phones today include these embedded accelerometers. The main contribution of this project work was to demonstrate how gait and activity recognition can be used on commercial mobile phones equipped with accelerometers. As stated before the main advantage of this method is that it provides unobtrusive user authentication and identification, and can be implemented on mobile phones. The proposed system in using activity recognition for gait recognition is applied to data from 45 volunteers. In order to analyze the data we have tried different approaches and algorithms for activity recognition and variants of existing methods used for gait. We did focus on the classification method to see which is performing better. We also implemented the methods for noise reduction algorithms, cycle detection approaches, averaging methods, applied in earlier works for gait recognition. The results are to be used in a real world development and is a next step towards a real gait and activity recognition application on either Android based phones or i-Phones.

Activity and gait recognition resulted in different performances and accuracies using several classification methods for the activity classification and a well-known method for the performance of gait recognition. For the activity classification, different test were performed and evaluated. As can be seen in the results for the classification, we further cannot conclude a best functioning approach, since most of the approaches are "winning" against each other when different test are tried out. By only looking at each persons classification of activities we mostly retrieve performances over 90% which is very useful. Here we can conclude that activity recognition, individual-based can be applied into applications. Furthermore, experimental results shows that in most cases that we can distinquish all subjects types of activities as well. This means that all subjects have special characteristics for either normal, fast slow in general. We observe this by the high accuricies beginning from approximately 70% and above.

Gait Recognition performance was calculated. By seperating normal, fast and slow walks for each user, we retrieved three overall equal error rates. What we observe is when the user is walking fast, then we retrieve best EER. This is due to the fact that a subject has more characteristics in the three directional movements. The performance of slow is worse, and this is due to the fact that we loose information at the template. The best EER gained was approximately 15% and could be further improved. However, this was not the main analysis in this project.

These results suggest the possibility of using the proposed method for protecting personal devices such as PDAs, mobile phones etc. In a future of truly pervasive computing, when small and inexpensive hardware can be embedded in various objects, this method could also be used for protecting valuable personal items. Moreover, reliably authenticated mobile devices may also serve as an automated authentication in relation to other systems such as access control system or automated external system logon.

Bibliography

[1] Krumm, J. & Horvitz, E. 2006. *Predestination: Inferring Destinations from Partial Trajectories*, volume 4206 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg.

[2] BenAbdelkader, C., Cutler, R., Nanda, H., & Davis, L. S. 2001. Eigengait: Motion-based recognition of people using image self-similarity. In *AVBPA*, 284–294.

[3] Larsen, P., Simonsen, E., & Lynnerup, N. 2008. Gait analysis in forensic medicine. *J Forensic Sci*.

[4] Nixon, M. S., Carter, J. N., Shutler, J., & Grant, M. 2002. New advances in automatic gait recognition. *Elsevier Information Security Technical Report*, 7(4), 23–35.

[5] Nixon, M. S., Tan, T. N., & Chellappa, R. 2005. *Human Identification based on Gait*. International Series on Biometrics. Springer.

[6] Niyogi, S. & Adelson, E. June 1994. Analyzing and recognizing walking figures in xyt. 469 –474.

[7] Wang, L., Tan, T., Hu, W., & Ning, H. 2003. Automatic gait recognition based on statistical shape analysis. *IEEE Transactions on Image Processing*, 12(9), 1120–1131.

[8] Sminchisescu, C., Kanaujia, A., Li, Z., & Metaxas, D. Learning to reconstruct 3d human motion from bayesian mixtures of experts. a probabilistic discriminative approach. Technical report, 2004.

[9] Bours, P. September 2006. *Authentication course, IMT 4721*. Gjøvik University College.

[10] Holien, K. 2008. Gait Recognition under non-standard circumstances, Master thesis, Gjøvik University College.

[11] Jain, A. K., Ross, A., & Prabhakar, S. 2004. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14, 4–20.

[12] Derawi, M. O. Multi-modal biometric authentication using gait and fingerprint on mobile devices. Master's thesis, Technical University of Danmark, 2009.

[13] Derawi, M. O., Bours, P., & Holien, K. 2010. Improved cycle detection for accelerometer based gait authentication. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing - Special Session on Advances in Biometrics*.

[14] TimesOnline. Aiming to catch criminals red-footed.

[15] CanadaNews. Forensic gait analysis points to accuse.

[16] BBC-News. How can you identify a criminal by the way they walk?

[17] Middleton, L., Buss, A. A., Bazin, A., & Nixon, M. S. 2005. A floor sensor system for gait recognition. In *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, 171–176, Washington, DC, USA. IEEE Computer Society.

[18] Bächlin, M., Schumm, J., Roggen, D., & Töster, G. 2009. Quantifying gait similarity: User authentication and real-world challenge. In *ICB '09: Proceedings of the Third International Conference on Advances in Biometrics*, 1040–1049, Berlin, Heidelberg. Springer-Verlag.

[19] Hynes, M., Wang, H., & Kilmartin, L. 2009. Off-the-shelf mobile handset environments for deploying accelerometer based gait and activity analysis algorithms. *Conf Proc IEEE Eng Med Biol Soc*, 1, 5187–90.

[20] Sprager, S. & Zazula, D. 2009. Gait identification using cumulants of accelerometer data. In *SENSIG'09/VIS'09/MATERIALS'09: Proceedings of the 2nd WSEAS International Conference on Sensors, and Signals and Visualization, Imaging and Simulation and Materials Science*, 94–99, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

[21] Henriksen, M., Lund, H., Moe-Nilssen, R., Bliddal, H., & Danneskiod-Samsře, B. 2004. Test-retest reliability of trunk accelerometric gait analysis. *Gait Posture*, 19(3), 288–97.

[22] Rong, L., Jianzhong, Z., Ming, L., & Xiang, H. 2005. Identification of individual walking patterns using gait acceleration. *In 1st International Conference in Bioinformatics and Biomedical Engineering*, 543–546.

[23] Rong, L., Jianzhong, Z., Ming, L., & Xiangfeng, H. 2007. A wearable acceleration sensor system for gait recognition. In *Industrial Electronics and Applications, 2007. ICIEA 2007*, 2654–2659.

[24] Gafurov, D. *Performance and Security Analysis of Gait-based User Authentication*. PhD thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, 2008.

[25] Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S.-M., & Ailisto, H. 2005. Identifying users of portable devices from gait pattern with accelerometers. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, 2, ii/973 – ii/976.

[26] Derawi, M. O. November 2010. Accelerometer-based gait analysis, a survey. *In Norwegian Information Security Conference (Norsk Informasjonsssikkerhetskonferanse, NISK*.

[27] Sazonov, E., Bumpus, T., Zeigler, S., & Marocco, S. 2005. Classification of plantar pressure and heel acceleration patterns using neural networks. In *International Joint Conference on Neural Networks, IJCNN '*. IEEE.

[28] Iso, T. & Yamazaki, K. 2006. Gait analyzer based on a cell phone with a single three-axis accelerometer. In *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, 141–144, New York, NY, USA. ACM.

[29] Mostayed, A., Kim, S., Mazumder, M. M. G., & Park, S. J. 2008. Foot step based person identification using histogram similarity and wavelet decomposition. In *ISA '08: Proceedings of the 2008 International Conference on Information Security and Assurance (isa 2008)*, 307–311, Washington, DC, USA. IEEE Computer Society.

[30] Annadhorai, A., Guenterberg, E., Barnes, J., Haraga, K., & Jafari, R. 2008. Human identification by gait analysis. In *HealthNet '08: Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, 1–3, New York, NY, USA. ACM.

[31] Ailisto, H. J., Lindholm, M., Mantyjarvi, J., Vildjiounaite, E., & Makela, S.-M. 2005. Identifying people from gait pattern with accelerometers. *In Proceedings of the SPIE*, 5779, 7–14.

[32] Ibrahim, R., Ambikairajah, E., Celler, B., Lovell, N., & Kilmartin, L. 2008. Gait patterns classification using spectral features. In *IET Irish Signals and Systems Conference (ISSC)*.

[33] Huang, B., Chen, M., Huang, P., & Xu, Y. 2007. Gait modeling for human identification. In *International Conference on Robotics and Automation*, 4833–4838.

[34] Morris, S. J. *A shoe-integrated sensor system for wireless gait analysis and real-time therapeutic feedback*. PhD thesis, Division of Health Sciences and Technology, Harvard University-MIT, 2004.

[35] Vildjiounaite, E., Mäkelä, S.-M., Lindholm, M., Riihimäki, R., Kyllönen, V., Mäntyjärvi, J., & Ailisto, H. 2006. Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices. In *Pervasive*, 187–201.

[36] Bours, P. & Shrestha, R. 5 2010. Eigensteps: A giant leap for gait recognition. In *2nd International Workshop on Security and Communication Networks (IWSCN 2010)*, Karlstad, Sweden.

[37] Huynh, D. T. G. *Human Activity Recognition with Wearable Sensors*. PhD thesis, TU Darmstadt, Darmstadt, September 2008.

[38] Wang, N., Ambikairajah, E., Lovell, N. H., & Celler, B. G. 2007. Accelerometry based classification of walking patterns using time-frequency analysis. *Conf Proc IEEE Eng Med Biol Soc*, 2007, 4899–902.

[39] Kwapisz, J. R., Weiss, G. M., & Moore, S. A. 2010. Activity recognition using cell phone accelerometers. *Human Factors*, 10–18.

[40] Mathie, M., Lovell, N., Coster, A., & Celler, B. 2002. Determining activity using a triaxial accelerometer. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society*

EMBS/BMES Conference, 2002. Proceedings of the Second Joint, volume 3, 2481 – 2482 vol.3.

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

[42] Lester, J., Choudhury, T., & Borriello, G. 2006. A Practical Approach to Recognizing Physical Activities. In *Pervasive Computing*, Fishkin, K., Schiele, B., Nixon, P., & Quigley, A., eds, volume 3968 of *Lecture Notes in Computer Science*, chapter 1, 1–16. Springer Berlin / Heidelberg, Berlin, Heidelberg.

[43] He, Z. & Jin, L. 2009. Activity recognition from acceleration data based on discrete consine transform and svm. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 5041 –5044.

[44] Streitz. Norbert, N. P. 2005. The disappearing computer. *Commun. ACM*, 32–35.

[45] Khan, A. M., Lee, Y. K., Lee, S. Y., & Kim, T. S. May 2010. Human Activity Recognition via an Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis. 1–6.

[46] Yang, J., Wang, S., Chen, N., Chen, X., & Shi, P. May 2010. Wearable accelerometer based extendable activity recognition system. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 3641 –3647.

[47] hee Lee, M., Kim, J., Kim, K., Lee, I., Jee, S. H., & Yoo, S. K. 2009. Physical activity recognition using a single tri-axis accelerometer. In *Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I, WCECS '09, October 20 - 22, 2009, San Francisco, USA*, Ao, S. I., Douglas, C., Grundfest, W. S., & Burgstone, J., eds, Lecture Notes in Engineering and Computer Science, 14–17. International Association of Engineers, Newswood Limited.

[48] Lester, J., Choudhury, T., Kern, N., Borriello, G., & Hannaford, B. 2005. A hybrid discriminative/generative approach for modeling human activities. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 766–772.

[49] Philipose, M., Fishkin, K., Perkowitz, M., Patterson, D., Fox, D., Kautz, H., & Hahnel, D. 2004. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, 3(4), 50 – 57.

[50] Stikic, M., Huynh, T., Van Laerhoven, K., & Schiele, B. January 2008. ADL Recognition Based on the Combination of RFID and Accelerometer Sensing. In *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health 2008)*, 258–263, Tampere, Finland. IEEE Xplore, IEEE Xplore.

[51] Ermes, M., Pärkka, J., Mantyjarvi, J., & Korhonen, I. January 2008. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 12(1), 20–26.

[52] Naeem, U., Bigham, J., & Wang, J. 2007. Recognising activities of daily life using hierarchical plans. In *Proceedings of the 2nd European conference on Smart sensing and context*, EuroSSC'07, 175–189, Berlin, Heidelberg. Springer-Verlag.

[53] Chen, J., Kam, A. H., Zhang, J., Liu, N., & Shue, L. 2005. Bathroom Activity Monitoring Based on Sound. 47–61.

[54] Tapia, E. M., Intille, S. S., & Larson, K. 2004. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Pervasive Computing*, Ferscha, A. & Mattern, F., eds, volume 3001 of *Lecture Notes in Computer Science*, 158–175–175. Springer Berlin / Heidelberg.

[55] Wang, S., Pentney, W., Popescu, A.-M., Choudhury, T., & Philipose, M. 2007. Common Sense Based Joint Training of Human Activity Recognizers. In *In: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2237–2242.

[56] Kern, N., Schiele, B., & Schmidt, A. 2003. Multi-Sensor Activity Context Detection for Wearable Computing. In *In Proc. EUSAI, LNCS*, volume 2875, 220–232.

[57] Mäntyjärvi, J., Himberg, J., & Seppänen, T. 2001. Recognizing human motion with multiple acceleration sensors. volume 2.

[58] Barralon, P., Noury, N., & Vuillerme, N. 2005. Classification of daily physical activities from a single kinematic sensor. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 3, 2447–2450.

[59] Bao, L. & Intille, S. S. 2004. Activity Recognition from User-Annotated Acceleration Data. *Pervasive Computing*, 1–17.

[60] Laerhoven, K. V. & Gellersen, H.-W. 2004. Spine versus porcupine: A study in distributed wearable activity recognition. In *Proceedings of the Eighth International Symposium on Wearable Computers*, ISWC '04, 142–149, Washington, DC, USA. IEEE Computer Society.

[61] Van Laerhoven, K. & Cakmakci, O. 2000. What shall we teach our pants? In *Wearable Computers, 2000. The Fourth International Symposium on*, 77 –83.

[62] Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. 2005. Activity Recognition from Accelerometer Data. *American Association for Artificial Intelligence*.

[63] Randell, C. & Muller, H. 2000. Context awareness by analysing accelerometer data. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, volume 0, 175–176, Los Alamitos, CA, USA. IEEE Comput. Soc.

[64] Choudhury, T. & Pentland, A. 2003. Sensing and modeling human networks using the sociometer. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ISWC '03, 216–, Washington, DC, USA. IEEE Computer Society.

[65] Bidargaddi, N., Klingbeil, L., & Sarela, A. 2007. Detecting walking activity in cardiac rehabilitation by using accelerometer. *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, 555–560.

[66] Ward, J. A., Lukowicz, P., & Tröster, G. 2005. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, 99–104, New York, NY, USA. ACM Press.

[67] Sekine, M., Tamura, T., Akay, M., Togawa, T., & Fukui, Y. 2000. Analysis of acceleration signals using wavelet transform. *Methods Inf Med*, 39(2), 183–5.

[68] Huynh, T., Blanke, U., & Schiele, B. september 2007. Scalable recognition of daily activities with wearable sensors. In *3rd International Workshop on Location- and Context- Awareness (LoCA 2007)*, 50–67.

[69] Nowozin, S., Bakir, G., & Tsuda, K. 2007. Discriminative subsequence mining for action classification. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1 –8.

[70] Huynh, T. & Schiele, B. May 2006. Unsupervised discovery of structure in activity data using multiple eigenspaces. In *2nd International Workshop on Location- and Context- Awareness (LoCA 2006)*, Dublin, Ireland.

[71] Patterson, D., Fox, D., Kautz, H., & Philipose, M. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, 44 – 51.

[72] Dunne, L., Walsh, P., Smyth, B., & Caulfield, B. 2006. Design and evaluation of a wearable optical sensor for monitoring seated spinal posture. *Wearable Computers, IEEE International Symposium*, 0, 65–68.

[73] Kunze, K., Barry, M., Heinz, E. A., Lukowicz, P., Majoe, D., & Gutknecht, J. 2006. Towards recognizing tai chi – an initial experiment using wearable sensors. In *Heinz1, Paul Lukowicz, Dennis Majoe, Jurg Gutknecht; Institute for Computer Systems and Networks UMIT*.

[74] Logan, B., Healey, J., Philipose, M., Tapia, E., & Intille, S. 2007. A Long-Term Evaluation of Sensing Modalities for Activity Recognition. In *UbiComp 2007: Ubiquitous Computing*, Krumm, J., Abowd, G. D., Seneviratne, A., & Strang, T., eds, volume 4717 of *Lecture Notes in Computer Science*, chapter 28, 483–500. Springer Berlin Heidelberg, Berlin, Heidelberg.

[75] Krause, A., München, T. U., & Smailagic, A. 2003. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *In Proceedings of the 7th International Symposium on Wearable Computers*, 88–97.

[76] Zhu, C. & Sheng, W. 2011. Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, PP(99), 1 –5.

[77] Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. 2006. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 113–116, Washington, DC, USA. IEEE Computer Society.

[78] Pirttikangas, S., Fujinami, K., & Nakajima, T. 2006. Feature selection and activity recognition from wearable sensors. In *UCS*, 516–527.

[79] Brady, S., Dunne, L. E., Tynan, R., Diamond, D., Smyth, B., & O'Hare, G. M. P. 2005. Garment-based monitoring of respiration rate using a foam pressure sensor. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, 214–215, Washington, DC, USA. IEEE Computer Society.

[80] Oliver, N. & Flores-Mangas, F. 2006. Healthgear: A real-time wearable system for monitoring and analyzing physiological signals. *Wearable and Implantable Body Sensor Networks, International Workshop on*, 0, 61–64.

[81] Keogh, E. J., Chu, S., Hart, D., & Pazzani, M. J. 2001. An online algorithm for segmenting time series. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, 289–296, Washington, DC, USA. IEEE Computer Society.

[82] Laerhoven, K. V. & Cakmakci, O. 2000. What shall we teach our pants? In *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, ISWC '00, 77–, Washington, DC, USA. IEEE Computer Society.

[83] Wang, S., Yang, J., Chen, N., Chen, X., & Zhang, Q. 2005. Human Activity Recognition with User-Free Accelerometers in the Sensor Networks. volume 2, 1212–1217.

[84] Yang, J.-Y., Wang, J.-S., & Chen, Y.-P. December 2008. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recogn. Lett.*, 29, 2213–2220.

[85] Ghasemzadeh, H., Loseu, V., Guenterberg, E., & Jafari, R. 2009. Sport training using body sensor networks: a statistical approach to measure wrist rotation for golf swing. In *Proceedings of the Fourth International Conference on Body Area Networks*, BodyNets '09, 2:1–2:8, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[86] Huynh, T. & Schiele, B. 2005. Analyzing features for activity recognition. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, 159–163, New York, NY, USA. ACM.

[87] Lombriser, C., Bharatula, N. B., Roggen, D., & Tröster, G. 2007. On-body activity recognition in a dynamic sensor network. In *Proceedings of the ICST 2nd international conference on Body area networks*, BodyNets '07, 17:1–17:6, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[88] Heinz, E. A., Kunze, K.-s., Sulistyo, S., Junker, H., Lukowicz, P., & Tröster, G. 2003. Experimental evaluation of variations in primary features used for accelerometric context recognition. In *in Proceedings of the 1st European Symposium on Ambient Intelligence (EUSAI 2003*, volume 1, 252–263.

[89] Chambers, G. S., Venkatesh, S., West, G. A. W., & Bui, H. H. 2002. Hierarchical recognition of intentional human gestures for sports video annotation. In *16th International Conference on Pattern Recognition*, volume 2, 1082–1085.

[90] Stikic, M., Van Laerhoven, K., & Schiele, B. September 2008. Exploring semi-supervised and active learning for activity recognition. In *Proceedings of the 12th International Symposium on Wearable Computers (ISWC 2008)*, 81–90, Pittsburgh, USA. IEEE Press, IEEE Press.

[91] Amft, O., Stäger, M., & Tröster, G. 2005. Analysis of chewing sounds for dietary monitoring. In *In UbiComp 2005*, 56–72. Springer.

[92] Lee, S. W. & Mase, K. July 2002. Activity and Location Recognition Using Wearable Sensors. *IEEE Pervasive Computing*, 1(3), 24–32.

[93] Ward, J. A., Lukowicz, P., Troster, G., & Starner, T. E. October 2006. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28, 1553–1567.

[94] Van Laerhoven, K., Kern, N., Gellersen, H.-w., & Schiele, B. 2003. Towards a wearable inertial sensor network. In *In Proceedings of IEE Eurowearable*, 125–130.

[95] Allen, F. R., Ambikairajah, E., Lovell, N. H., & Celler, B. G. August 2006. An Adapted Gaussian Mixture Model Approach to Accelerometry-Based Movement Classification Using Time-Domain Features. 3600–3603.

[96] Heinz, E. A., Kunze, K. S., Gruber, M., Bannach, D., & Lukowicz, P. 2006. Using Wearable Sensors for Real-Time Recognition Tasks in Games of Martial Arts - An Initial Experiment. *Computational Intelligence and Games, 2006 IEEE Symposium on*, 98–102.

[97] Liu, B. January 2009. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer, 1st ed. 2007. corr. 2nd printing edition.

[98] Huynh, T. & Schiele, B. 2006. Towards less supervision in activity recognition from wearable sensors. In *Wearable Computers, 2006 10th IEEE International Symposium on*, 3 –10.

[99] Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*.

[100] Carey, Michael J., C. S. 2006. *Data-Centric Systems and Applications*.

[101] Minnen, D., Starner, T., Essa, M., & Isbell, C. 2006. Discovering Characteristic Actions from On-Body Sensor Data. *Wearable Computers, 2006 10th IEEE International Symposium on*, 11–18.

[102] Wyatt, D., Philipose, M., & Choudhury, T. 2005. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, 21–27. AAAI Press.

[103] Nguyen, A., Moore, D., & McCowan, I. 2007. Unsupervised clustering of free-living human activities using ambulatory accelerometry. *Conf Proc IEEE Eng Med Biol Soc*, 2007, 4895–8.

[104] Clarkson, B. & Pentland, A. 1999. Unsupervised clustering of ambulatory audio and video. In *ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference*, volume 6, 3037–3040, Washington, DC, USA. IEEE Computer Society.

[105] Patterson, D. J., Liao, L., Gajos, K., Collier, M., Livic, N., Olson, K., Wang, S., Fox, D., & Kautz, H. 2004. Opportunity Knocks: A System to Provide Cognitive Assistance with Transportation Services. 433–450.

[106] Mahajan, D., Kwatra, N., Jain, S., Kalra, P., & Banerjee, S. December 2004. A framework for Activity Recognition and Detection of Unusual Activities. In *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*.

[107] Machine learning - who's the boss? http://awesomeful.net/posts/78-machine-learning-who-s-the-boss. [Online; accessed 13-march-2011].

[108] Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. 2006. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 113–116, Washington, DC, USA. IEEE Computer Society.

[109] Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. 2006. Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1), 156–167.

[110] Derawi, M. O., Nickel, C., Bours, P., & Busch, C. 2010. Unobtrusive user-authentication on mobile phones using biometric gait recognition. *Intelligent Information Hiding and Multimedia Signal Processing, International Conference on*, 0, 306–311.

[111] Reutemann, D. S. P. *WEKA Experimenter Tutorial for Version 3-5-5*. David Scuse and University of Waikato, January 26 2007.

A Participant Agreement Declaration

Participant Agreement Declaration

Participation in acquisition of gait data

I am participating in the acquisition of gait data on a voluntarily basis. The data are taken using accelerometers in mobile devices to fulfill the purpose that is described in detail on the back side of this sheet.

The data processing institutions are the Gjøvik University College (Høgskolen i Gjøvik). These institutions take care that the recorded data are solely used for teaching and research purposes.

With my signature I confirm the following:

1.) I have been informed in oral and written form about the content and purpose of the collected data that is in relation to my person.
2.) My data will only be used to serve this purpose. The detailed description of the purpose is documented on the back side of this sheet.
3.) I allow that gait data from me are collected.
4.) I have been informed that I can reject to sign the agreement.
5.) I have been informed that I can request to receive insight in the collected data before such data is used for teaching and research purposes.
6.) I know that I can withdraw my participation anytime I want without giving any explanation and all data collected from we will be deleted permanently.

All data will be deleted respectively the link between the data and my name will be destroyed as soon as it is not necessary to maintain it. This will happen as the research experiment has been completed.

First name - family name: ________________________________

<u>Meta information:</u>

Gender: ☐ male ☐ female

Age __________

Height (in cm) _____________ Weight (in kg) __________

Length of leg (in cm) __________

Kind of worn shoes ____

Time __________

Gjøvik, date _________________ signature: _____________________

Background information for this agreement

Purpose of this project
From each participant we capture video data while the participant is walking.

The data will be used for the following purpose:
Provision of data to the biometric research groups in the Gjøvik University College. The data will be stored and analyzed without link to the name of the student but with the research relevant meta data such as age and gender.

Background information

The recorded data will be used to develop and test methods which allow the authentication on mobile devices by gait recognition. Data on mobile devices is often insufficiently protected as it is more comfortable for the user to stay logged-in. This means that anybody having physical access to the mobile device can directly access all data stored in it. This shows the need for user-friendly authentication methods which enable an unobtrusive authentication.

A second focus is on technology research for enabling privacy protection of the stored references (so called biometric templates) in a biometric system. Since biometric technologies are widely adopted in multiple applications, the threat of compromising the biometric templates becomes ever more serious. Based on earlier studies it is expected that this research will lead to new technological measures that allows for templates that prevent the possibility for cross-matching and associated data mining, and allows for renewability in case the biometric record is compromised.
The data set will be used to validate the scientific research results. It will be taken care that no attempt is being made to analyze the captured signals regarding diseases or personal characteristics and habits of the subject. For the purpose of this research project it is only of relevance, whether or not recognition can be performed with high reliability.

B Tables

All the numbers in this Appendix indicates the classification percentage (%) accuricies with its classification approcahes used and a small descriptive caption.

Table 38: Personal Cross Validation with 4 features

	BayesNet	**NaiveBayes**	**LibSVM**	**MultilayerPercepton**	**RBFNetwork**	**RandomTree**	**LMT**
Nr							
1	96.8354	98.1013	96.2025	98.1013	98.1013	96.2025	98.1013
2	92.0863	89.2086	85.6115	89.9281	91.3669	92.0863	92.8058
3	98.0198	96.5347	95.0495	97.0297	97.0297	98.5149	98.5149
4	95.1515	93.3333	96.3636	95.7576	96.9697	98.7879	98.7879
5	97.973	91.8919	95.2703	96.6216	93.9189	95.2703	98.6486
6	84.6626	80.3681	80.3681	87.1166	87.1166	78.5276	82.8221
7	98.6364	97.7273	96.3636	99.5455	98.1818	96.8182	99.5455
8	94.8276	93.1034	93.6782	94.8276	96.5517	93.6782	94.2529
9	95.1613	94.086	94.6237	97.8495	96.7742	95.1613	98.3871
10	97.7169	92.2374	96.347	97.2603	96.8037	93.6073	96.8037
11	83.871	70.5069	85.2535	76.9585	72.3502	86.1751	87.0968
12	91.9192	99.4949	95.4545	98.4848	97.4747	93.4343	98.4848
13	95.5414	96.1783	91.0828	96.8153	98.0892	95.5414	98.7261
14	98.8571	100	94.8571	100	100	99.4286	99.4286
15	97.561	90.2439	95.6098	91.2195	91.2195	94.1463	96.5854
16	97.549	97.549	95.5882	97.549	97.549	94.1176	97.549
17	95.0355	87.234	92.1986	87.234	90.0709	95.0355	99.2908
18	86.9281	84.3137	94.1176	91.5033	90.8497	90.8497	95.4248
19	96.7105	97.3684	96.0526	98.6842	96.7105	92.7632	97.3684
20	97.9592	96.9388	94.898	98.4694	98.4694	96.9388	98.4694
21	87.5648	60.6218	83.4197	82.3834	81.3472	88.601	89.1192
22	99.4475	85.0829	94.4751	91.7127	91.7127	97.2376	99.4475
23	95.4286	96	95.4286	97.1429	97.7143	94.2857	96.5714
24	97.1591	93.75	92.0455	97.7273	96.0227	96.0227	94.8864

Table 39: Personal Cross Validation with 4 features- Continued

	BayesNet	**NaiveBayes**	**LibSVM**	**MultilayerPercepton**	**RBFNetwork**	**RandomTree**	**LMT**
25	99.375	96.875	94.375	98.125	97.5	96.875	95
26	98.0676	94.686	97.5845	99.5169	97.1014	99.5169	100
27	87.6923	70.7692	77.4359	74.359	71.7949	83.0769	85.641
28	90.0763	75.5725	87.7863	95.4198	80.916	91.6031	92.3664
29	93.5135	91.3514	96.2162	98.9189	95.1351	95.1351	97.8378
30	97.546	95.092	90.7975	96.9325	94.4785	95.092	97.546
31	98.9474	95.2632	93.6842	97.8947	96.3158	97.3684	95.7895
32	90.411	89.0411	88.3562	89.726	92.4658	89.0411	91.0959
33	99.435	93.2203	92.6554	97.1751	94.3503	92.6554	96.6102
34	92.638	97.546	94.4785	99.3865	97.546	96.9325	99.3865
35	96.2963	97.5309	97.5309	95.679	95.679	96.9136	95.0617
36	90.9548	89.4472	89.9497	91.9598	88.4422	91.9598	93.4673
37	96.3731	91.1917	91.1917	92.7461	94.3005	92.228	95.3368
38	97.8261	82.6087	97.2826	83.1522	84.2391	93.4783	98.3696
39	95	70.5	89	73	80	94	97.5
40	94.8052	42.8571	92.2078	56.4935	57.1429	89.6104	100
41	95	90	93.5714	92.8571	91.4286	93.5714	95.7143
42	99.4565	100	94.5652	99.4565	100	97.8261	99.4565
43	99.5169	98.5507	98.0676	100	99.5169	96.6184	99.5169
44	87.4251	78.4431	87.4251	82.6347	80.8383	88.024	92.2156
45	96.6851	96.6851	92.2652	99.4475	97.7901	99.4475	98.895
AVG	**94.8809778**	**89.313462**	**92.595256**	**92.77339778**	**91.9861311**	**93.87124222**	**96.08723**

Table 40: Personal Traning Testing 3-12 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
1	92.562	98.3471	95.8678
2	65.7658	58.5586	54.0541
3	61.7284	48.1481	57.4074
4	64.3939	59.0909	57.5758
5	57.5	55	50
6	59.8485	56.0606	55.303
7	70.9677	89.7849	76.3441
8	84.5588	75.7353	77.2059
9	73.5099	91.3907	91.3907
10	78.1609	61.4943	68.3908
11	43.956	38.4615	43.4066
12	72.2581	76.129	81.9355
13	77.6978	74.8201	69.0647
14	92.5926	95.5556	83.7037
15	78.0488	63.4146	53.0488
16	80.7453	86.3354	83.2298
17	45.5357	45.5357	55.3571
18	41.7323	49.6063	53.5433
19	95.3125	78.9063	80.4688
20	63.1902	76.0736	72.3926
21	32.9114	36.7089	41.1392
22	51.7007	33.3333	42.1769
23	92.3611	95.8333	73.6111
24	70	77.8571	66.4286
25	93.0769	71.5385	70.7692
26	98.2249	100	100
27	36.1446	19.2771	21.0843
28	58.8785	57.0093	50.4673
29	49.3333	60.6667	60
30	90	81.4286	84.2857
31	85.443	84.1772	86.7089
32	67.2566	55.7522	53.0973
33	74	43.3333	36
34	62.8571	80	75.7143
35	92.9688	90.625	86.7188
36	38.3117	35.7143	39.6104
37	53.125	26.25	36.25
38	60.9272	52.3179	45.0331
39	37.3418	26.5823	29.1139
40	27.907	29.4574	40.3101
41	88.7931	86.2069	82.7586
42	91.7241	97.2414	91.7241
43	96.9697	78.7879	73.3333
44	63.6364	57.5758	64.3939
45	83.4437	92.7152	94.0397
Ang	**68.83115111**	**65.52973778**	**64.543538**

Table 41: Personal Traning Testing 6-9 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
1	96.5909	85.2273	90.9091
2	89.5349	88.3721	84.8837
3	79.6748	65.8537	63.4146
4	72.7273	88.8889	90.9091
5	90.5263	90.5263	91.5789
6	51.0638	68.0851	70.2128
7	89.7059	99.2647	92.6471
8	82.2917	80.2083	90.625
9	91.8182	86.3636	84.5455
10	89.9225	82.9457	82.1705
11	53.9007	50.3546	32.6241
12	92.562	77.686	80.9917
13	81.3725	89.2157	74.5098
14	90.625	94.7917	92.7083
15	86.7769	63.6364	63.6364
16	86.4	94.4	92
17	44.7368	38.1579	36.8421
18	57	70	60
19	93	88	88
20	76.5625	70.3125	64.8438
21	7.7586	5.1734	25
22	77.193	84.2105	77.193
23	88.5714	95.2381	89.5238
24	82.8283	72.7273	66.6667
25	87.7778	90	86.6667
26	97.5806	100	100
27	60.8	62.4	41.6
28	61.25	81.25	75
29	82.1429	90.1786	70.5357
30	75.7576	86.8687	93.9394
31	84.8	96	90.4
32	71.0843	72.2892	65.0602
33	87.5	95.5357	96.4286
34	86.2385	99.0826	96.3303
35	95.8763	87.6289	87.6289
36	86.4407	84.7458	71.1864
37	66.3934	93.4426	94.2623
38	84.9558	78.7611	76.1062
39	40.678	62.7119	22.8814
40	20.4301	7.5269	7.5269
41	88.2353	90.5882	92.9412
42	98.1132	97.1698	94.3396
43	95.8678	99.1736	92.562
44	60.4167	48.9583	45.8333
45	95.7265	100	99.1453
Avg	**77.36021111**	**78.97670444**	**75.2624533**

Table 42: Personal Traning Testing 9-6 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
1	98.3333	96.6667	100
2	83.6066	77.0492	86.8852
3	69.1358	70.3704	75.3086
4	73.913	86.9565	79.7101
5	41.3043	78.2609	23.913
6	57.8125	60.9375	51.5625
7	96.6292	98.8764	98.8764
8	59.375	51.5625	51.5625
9	98.7013	96.1039	97.4026
10	82.6087	91.3043	85.8696
11	66.2921	59.5506	43.8202
12	98.7179	100	100
13	72.0588	88.2353	86.7647
14	100	100	100
15	82.8947	75	86.8421
16	96.6667	95.556	93.3333
17	55.5556	44.4444	68.8889
18	55.7377	55.7377	40.9836
19	96.9697	95.4545	98.4848
20	87.6443	92.5926	91.358
21	33.3333	50	50
22	73.3333	76	64
23	88.4058	92.7536	75.3623
24	87.5	93.75	84.375
25	68.9655	86.2069	79.3103
26	100	100	100
27	69.7674	58.1395	59.3023
28	44.2623	77.0492	72.1311
29	87.5	91.6667	91.6667
30	87.5	95.8333	95.8333
31	84.9315	86.3014	98.6301
32	84.7458	62.7119	54.2373
33	82.1918	98.6301	95.8904
34	93.5897	98.7179	96.1538
35	81.4286	82.8571	82.8571
36	96.5517	88.5057	96.5517
37	96.1538	100	100
38	84.6154	76.9231	79.4872
39	64.1026	50	44.8718
40	13.2075	0	1.8868
41	76.4706	94.1176	86.2745
42	100	96.5517	96.5517
43	95.0617	100	98.7654
44	61.1111	75	73.6111
45	97.5	100	98.75
Avg	**78.35970222**	**81.03055778**	**78.623689**

Table 43: Global Cross Validation with 4 features

	BayesNet	**NaiveBayes**	**LibSVM**	**MultilayerPercepton**	**RBFNetwork**	**RandomTree**	**LMT**
Nr							
	73.6255	**71.5717**	**79.5867**	**73.1371**	**72.8741**	**75.1659**	**79.624**

Table 44: Session Individual 3-12 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
1	81.8182	66.9421	80.1653
2	81.0811	89.1892	69.3694
3	51.8519	48.7654	68.5185
4	55.303	49.2424	63.6364
5	80	89.1667	66.6667
6	53.3333	60.6061	56.8182
7	52.6882	48.3871	61.828
8	62.5	71.3235	50.7353
9	71.1854	67.5497	68.8742
10	32.1839	45.977	51.1494
11	54.1429	61.5385	47.2527
12	49.6774	63.2258	58.7097
13	94.2446	98.5612	84.8921
14	70.3704	48.8889	55.5556
15	53.0488	52.439	53.0488
16	86.3354	59.6273	63.354
17	57.1429	73.2143	57.1429
18	63.7795	53.5433	54.3307
19	69.5313	83.5938	74.2188
20	87.7301	66.8712	80.3681
21	65.1899	61.3924	54.4304
22	71.4286	75.5102	69.3878
23	85.4167	89.5833	75
24	55.7143	57.1429	48.5714
25	64.6154	83.0769	74.6154
26	88.1657	98.8166	83.432
27	63.253	56.6265	48.7952
28	52.3364	73.8318	63.5514
29	76.6667	75.3333	52.6667
30	85	82.8571	75
31	67.7215	70.8861	74.6835
32	73.4513	77.8761	65.4867
33	93.3333	89.3333	82.6667
34	81.4286	87.1429	73.5714
35	69.5313	42.1875	52.3438
36	83.7662	96.7532	98.7013
37	76.25	91.25	64.375
38	52.9801	52.3179	52.3179
39	68.9873	66.4557	69.6203
40	19.3798	19.3798	21.7054
41	58.6207	63.7931	69.8276
42	88.9655	99.3103	84.8276
43	89.697	89.697	86.6667
44	65.9091	67.4242	73.4848
45	58.9404	58.2781	41.0596
Avg	**68.10438**	**69.44241556**	**64.965**

Table 45: Session Individual 6-9 with 4 features

	BayesNet	MultilayerPercepton	LMT
Nr			
1	77.2727	51.1364	38.6364
2	80.2326	82.5581	70.9302
3	52.0325	58.5366	69.1057
4	53.5354	57.5758	58.5859
5	95.7895	94.7368	84.2105
6	60.6383	54.2553	57.4468
7	53.6765	56.6176	60.2941
8	53.125	66.6667	73.9583
9	65.4545	75.4545	80
10	32.5581	62.7907	46.5116
11	53.9007	77.305	59.5745
12	61.157	62.8099	70.2479
13	93.1373	97.0588	85.2941
14	33.3333	44.7917	43.75
15	53.719	46.281	52.0661
16	72	50.4	92.8
17	64.4737	67.1053	40.7895
18	47	46	40
19	83	78	79
20	89.0625	87.5	79.6875
21	61.2069	65.5172	43.1034
22	78.9474	75.4386	71.9298
23	80.9524	71.4286	62.8571
24	47.4747	51.5152	33.3333
25	56.6667	60	62.2222
26	96.7742	100	94.3548
27	67.2	59.2	63.2
28	55	70	77.5
29	67.8571	72.3214	65.1786
30	83.8384	77.7778	77.7778
31	88.8	79.2	68.8
32	68.6747	62.6506	57.8313
33	96.4286	97.3214	95.5357
34	71.5596	69.7248	77.9817
35	80.4124	63.9175	90.7216
36	99.1525	98.3051	86.4407
37	87.7049	84.4262	63.9344
38	58.4071	30.9735	58.4071
39	64.4068	57.6271	53.3898
40	9.6774	18.2796	21.5054
41	74.1176	80	67.0588
42	99.0566	99.0566	95.283
43	87.6033	100	95.8678
44	73.9583	68.75	64.5833
45	46.1538	48.7179	66.6667
Avg	**68.380667**	**68.43842889**	**66.6301**

Table 46: Session Based 3-12 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
Avg	**68.284**	**69.1929**	**64.9723**

Table 47: Session Based 6-9 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
Avg	**68.5655**	**68.7719**	**67.162**

Table 48: Session Based 9-6 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
Avg	**68.1182**	**67.3717**	**67.6205**

Table 49: User Based 5-40 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Avg	**63.502**	**65.6071**	**61.132**

Table 50: User Based 10-35 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Avg	**64.0316**	**67.3161**	**64.0316**

Table 51: User Based 15-30 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Avg	**67.5984**	**62.9017**	**63.5102**

Table 52: User Based 20-25 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Avg	**69.9977**	**70.4509**	**64.5139**

Table 53: User Individual 5-40 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
6	57.0552	65.6442	60.1227
7	62.7273	42.7273	58.1818
8	72.4138	78.1609	54.023
9	77.4194	65.0538	39.7849
10	78.0822	46.1187	54.3379
11	46.5438	59.9078	35.4839
12	69.697	10.6061	58.0808
13	84.0764	91.7197	85.9873
14	49.1429	52.5714	32.5714
15	58.5366	60.4878	53.6585
16	77.451	69.1176	75
17	53.9007	72.3404	58.8652
18	68.6275	70.5882	61.4379
19	65.7895	71.7105	76.3158
20	59.1837	70.9184	66.8367
21	44.0415	57.513	52.8497
22	77.3481	56.3536	69.0608
23	66.8571	96.5714	71.4286
24	52	58.5227	48.8636
25	54.375	70.625	63.75
26	67.6329	87.9227	63.7681
27	63.0769	47.1795	60
28	43.5115	69.4656	53.4351
29	72.4324	91.3514	65.9459
30	58.2822	90.184	63.8037
31	75.2632	36.8421	64.7368
32	84.9315	84.9315	74.6575
33	62.1469	81.3559	73.4463
34	62.5767	80.3681	58.8957
35	54.321	49.3827	43.8272
36	72.3618	84.4221	85.4271
37	70.4663	84.456	87.0466
38	50	32.6087	47.8261
39	41.5	67.5	51
40	44.1558	27.9221	40.2597
41	72.1429	69.2857	52.1429
42	80.9783	98.913	96.7391
43	79.7101	86.9565	63.285
44	47.3054	66.4671	59.8802
45	53.5912	38.674	63.5359
Avg	**63.29821**	**66.08618**	**61.15749**

Table 54: User Individual 10-35 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
11	52.0737	64.0553	64.977
12	73.2323	84.3434	70.202
13	85.3503	93.6306	80.2548
14	61.7143	65.1429	68
15	44.878	41.9512	49.7561
16	78.9216	67.6471	64.2157
17	44.6809	51.0638	53.1915
18	60.1307	52.9412	41.1765
19	63.8158	65.1316	47.3684
20	76.0204	92.8571	85.7143
21	39.3782	40.4145	50.2591
22	76.2431	80.663	72.9282
23	50.2857	44.5714	55.4286
24	57.9545	68.1818	57.3864
25	53.125	40.625	60.625
26	77.2947	81.1594	79.7101
27	69.2308	69.7436	53.8462
28	50.3817	67.1756	46.5649
29	64	71.3514	54.0541
30	71.1656	84.0491	73.0061
31	83.1579	89.4737	76.8421
32	71.9178	58.2192	74.6575
33	79.661	79.661	75.1412
34	69.3252	68.0982	74.8466
35	72.2222	95.679	65.4321
36	74.8744	89.4472	77.8894
37	48.1865	39.3782	59.0674
38	40.2174	49.4565	41.3043
39	55	56	53.5
40	44.1558	38.3117	39.6104
41	55.7143	66.4286	69.2857
42	91.3043	94.0217	98.3696
43	92.7536	89.372	73.913
44	48.503	55.6886	74.8503
45	50.8287	47.5138	45.3039
Avg	**63.642377**	**66.95566857**	**63.6765**

Table 55: User Individual 20-25 with 4 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
21	49.2228	38.342	47.1503
22	74.5856	76.7956	73.4807
23	57.7143	49.7143	48.5714
24	76.1364	80.6818	72.1591
25	56.25	45	59.375
26	87.9227	88.4058	66.1836
27	68.2051	70.7692	59.4872
28	70.9924	73.2824	44.2748
29	83.7838	80	58.9189
30	71.7791	85.2761	67.4847
31	84.7368	96.3158	87.8947
32	82.8767	67.1233	65.0685
33	88.7006	85.8757	78.5311
34	77.3006	76.0736	69.3252
35	72.2222	96.2963	80.2469
36	89.4472	89.4472	89.4472
37	50.2591	44.0415	53.3679
38	45.1087	50.5435	46.1957
39	63	59.5	57.5
40	33.7662	39.6104	37.013
41	52.8571	64.2857	60
42	97.2826	95.6522	91.8478
43	85.9903	90.8213	69.0821
44	55.6886	58.0838	65.2695
45	62.4309	50.2762	54.6961
Avg	**69.510392**	**70.088548**	**64.1029**

Table 56: Personal Cross Validation with 8 features

	BayesNet	NaiveBayes	LibSVM	MultilayerPercepton	RBFNetwork	RandomTree	LMT
Nr							
1	94.9367	97.4684	39.8734	97.4684	96.8354	95.5696	99.3671
2	93.5252	89.9281	31.6547	92.8058	92.8058	92.8058	89.2086
3	96.5347	95.5446	38.6139	97.5248	95.0495	97.0297	99.0099
4	96.9697	93.9394	43.0303	98.7879	94.5455	97.5758	96.3636
5	89.8649	90.5405	38.5135	97.973	88.5135	92.5676	95.9459
6	84.6626	80.9816	38.0368	89.5706	84.0491	82.8221	86.5031
7	98.1818	95.4545	39.5455	100	96.3636	97.7273	99.5455
8	95.977	85.0575	47.7011	94.8276	91.3793	93.1034	95.4023
9	95.6989	90.8602	43.0108	98.3871	94.086	96.2366	97.3118
10	97.2603	92.2374	45.6621	96.8037	94.9772	94.0639	97.2603
11	83.4101	69.5853	37.3272	79.7235	70.0461	82.9493	91.2442
12	92.4242	97.4747	48.9899	98.4848	94.4444	95.9596	97.9798
13	96.1783	96.8153	36.9427	97.4522	94.9045	95.5414	98.0892
14	99.4286	100	37.7143	100	100	98.2857	99.4286
15	97.0732	88.2927	N/A	95.122	94.1463	93.6585	95.6098
16	99.5098	98.0392	43.1373	98.5294	99.0196	96.0784	96.0784
17	92.9078	87.234	41.1348	87.9433	89.3617	90.7801	97.1631
18	89.5425	77.1242	34.6405	94.7712	91.5033	90.1961	94.7712
19	98.0263	94.7368	38.8158	98.0263	95.3947	95.3947	98.0263
20	97.9592	96.9388	38.2653	98.4694	97.449	96.9388	97.9592
21	89.6373	64.7668	35.7513	84.5285	74.6114	87.0466	88.0829
22	98.895	85.6354	40.3315	90.6077	83.9779	95.5801	99.4475
23	97.7143	96.5714	41.1429	97.7143	97.7143	96	97.7143
24	97.7273	93.75	46.5909	96.5909	97.1591	94.8864	96.5909

Table 57: Personal Cross Validation with 8 features

	BayesNet	**NaiveBayes**	**LibSVM**	**MultilayerPercepton**	**RBFNetwork**	**RandomTree**	**LMT**
Nr							
25	100	95.625	36.25	99.375	97.5	96.25	98.125
26	98.0676	94.686	44.4444	98.0676	96.1353	99.0338	100
27	81.0256	71.7949	38.9744	73.8462	74.8718	82.5641	84.1026
28	86.2595	67.9389	45.8015	94.6565	80.1527	85.4962	93.1298
29	99.4595	92.4324	36.2162	100	97.8378	97.2973	99.4595
30	95.092	91.411	48.4663	97.546	92.0245	94.4785	96.319
31	97.3684	91.0526	45.7895	97.8947	96.8421	97.8947	96.3158
32	93.1507	82.1918	37.6712	91.0959	86.3014	91.7808	91.0959
33	98.3051	92.6554	45.1977	96.6102	94.3503	92.6554	97.7401
34	93.865	95.7055	41.1043	98.773	91.411	97.546	99.3865
35	96.9136	96.2963	43.2099	98.1481	96.2963	96.2963	98.7654
36	90.4523	89.4472	38.191	94.4724	86.9347	92.4623	90.9548
37	95.8549	91.1917	39.8964	94.8187	93.2642	94.8187	95.3368
38	96.1957	82.0652	38.0435	86.413	79.8913	96.1957	98.3696
39	87	71	34.5	70	71.5	88.5	98.5
40	92.8571	43.5065	37.6623	64.2857	59.0909	91.5584	98.7013
41	95.7143	90	39.2857	97.1429	89.2857	95.7143	96.4286
42	100	100	42.9348	100	100	100	100
43	99.5169	98.0676	40.0966	100	99.0338	96.6184	99.0338
44	86.8263	73.6527	N/A	82.0359	76.0479	87.4251	94.6108
45	96.1326	96.6851	N/A	99.4475	98.3425	99.4475	100
AVG	**94.5356178**	**88.141836**	**N/A**	**93.70537111**	**90.3433644**	**93.84068889**	**96.23286**

Table 58: Global Cross Validation with 8 features

	BayesNet	**NaiveBayes**	**LibSVM**	**MultilayerPercepton**	**RBFNetwork**	**RandomTree**	**LMT**
Avg	**72.9493**	**72.2981**	**NA**	**75.4164**	**72.7113**	**76.3306**	**80.2129**

Table 59: Session Individual 3-12 with 8 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
1	69.4215	78.5124	77.686
2	82.8829	85.5856	82.8829
3	51.8519	46.2963	69.7531
4	49.2424	47.7273	63.6364
5	90.8333	95	78.3333
6	65.1515	57.5758	61.3636
7	47.3118	51.0753	53.7634
8	63.9706	71.3235	47.0588
9	75.4967	68.2119	70.1987
10	51.1494	55.1724	57.4713
11	58.7912	64.8352	56.044
12	56.129	57.4194	57.4194
13	96.4029	98.5612	90.6475
14	71.8519	54.8148	52.5926
15	57.3171	61.5854	55.4878
16	46.5839	73.913	53.4161
17	68.75	75	56.25
18	71.6535	62.9921	51.1811
19	61.7188	85.9375	85.1563
20	79.1411	87.1166	79.7546
21	60.7595	61.3924	58.2278
22	66.6667	75.5102	68.0272
23	78.4722	82.6389	72.2222
24	55.7143	54.2857	57.1429
25	70	80	80
26	96.4497	100	91.716
27	64.4578	60.8434	37.3494
28	56.0748	68.2243	71.028
29	75.3333	74.6667	67.3333
30	79.2857	83.5714	79.2857
31	71.519	85.443	65.8228
32	74.3363	78.7611	75.2212
33	94	98	80
34	75	75.7143	85.7143
35	85.9375	51.5625	64.8438
36	97.4026	96.7532	98.7013
37	75.625	80	63.125
38	50.9934	42.3841	39.0728
39	68.9873	67.7215	70.2532
40	20.155	20.155	18.6047
41	85.3448	81.0345	72.4138
42	94.4828	99.3103	97.931
43	90.303	98.7879	86.0606
44	66.6667	65.1515	71.2121
45	37.7483	56.2914	45.0331
Avg	**69.052602**	**71.48575556**	**67.03198**

Table 60: Session Individual 6-9 with 8 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
1	50	57.9545	63.6364
2	90.6977	82.5581	81.3953
3	59.3496	65.0407	65.0407
4	48.4848	58.5859	63.6364
5	88.4211	91.5789	91.5789
6	69.1489	56.383	61.7021
7	48.5294	54.4118	50.7353
8	47.9167	65.625	71.875
9	67.2727	71.8182	68.1818
10	70.5426	62.7907	58.1395
11	53.9007	68.0851	55.3191
12	61.157	66.1157	66.1157
13	94.1176	96.0784	89.2157
14	53.125	48.9583	52.0833
15	51.2397	62.8099	53.719
16	48.8	87.2	59.2
17	64.4737	67.1053	50
18	54	56	49
19	73	89	74
20	82.0313	85.9375	77.3438
21	63.7931	52.5862	45.6897
22	74.5614	77.193	72.807
23	72.381	67.619	65.7143
24	51.5152	57.5758	56.5657
25	64.4444	62.2222	73.3333
26	99.1935	100	96.7742
27	64	61.6	57.6
28	47.5	68.75	70
29	76.7857	70.5357	75
30	80.8081	81.8182	80.8081
31	92	86.4	75.2
32	62.6506	75.9036	66.2651
33	91.0714	98.2143	85.7143
34	68.8073	65.1376	72.4771
35	94.8454	70.1031	79.3814
36	98.3051	99.1525	89.8305
37	77.8689	80.3279	74.4098
38	56.6372	38.0531	62.8319
39	60.1695	65.2542	53.3898
40	9.6774	16.129	17.2043
41	75.2941	80	87.0588
42	97.1698	100	93.3962
43	89.2562	100	96.6942
44	81.25	70.8333	67.7083
45	45.2991	52.1368	35.8974
Avg	**68.25539778**	**70.92405556**	**67.85932**

Table 61: Session Based 3-12 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	**68.7616**	**71.3339**	**66.7591**

Table 62: Session Based 6-9 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	**68.483**	**71.2487**	**67.8019**

Table 63: Session Based 9-6 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	**68.8647**	**70.7932**	**69.6734**

Table 64: User Based 5-40 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	67.782	57.8698	57.9813

Table 65: User Based 10-35 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	65.4448	62.4698	58.9438

Table 66: User Based 15-30 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	68.167	63.5672	62.7496

Table 67: User Based 20-25 with 8 features

	BayesNet	MultilayerPercepton	LMT
Avg	68.1622	72.717	59.8459

Table 68: User Individual 5-40 with 8 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
6	60.7362	53.3742	63.1902
7	67.7273	62.2727	57.7273
8	58.046	77.0115	58.046
9	79.5699	56.9892	55.3763
10	84.4749	45.6621	36.5297
11	51.1521	59.9078	47.0046
12	78.7879	18.1818	18.1818
13	91.7197	89.8089	83.4395
14	58.8571	17.1429	29.7143
15	45.8537	52.6829	53.1707
16	71.5686	47.549	71.0784
17	56.7376	67.3759	62.4113
18	60.1307	58.8235	54.902
19	79.6053	64.4737	73.0263
20	69.898	71.9388	70.4082
21	47.6684	50.2591	50.7772
22	71.2707	60.221	45.3039
23	60.5714	79.4286	65.1429
24	65	60.2273	48.8636
25	55.625	56.25	58.75
26	84.058	65.7005	66.1836
27	68.2051	39.4872	54.359
28	44.2748	67.1756	54.1985
29	82.7027	64.8649	62.7027
30	67.4847	80.3681	69.3252
31	83.6842	35.2632	43.1579
32	83.5616	82.1918	84.9315
33	73.4463	64.4068	61.5819
34	64.4172	52.1472	52.7607
35	79.0123	29.0123	61.1111
36	84.4221	78.8945	80.402
37	52.3316	62.6943	81.8653
38	38.0435	23.3696	31.5217
39	58.5	50	46
40	44.8052	40.2597	44.1558
41	60	62.1429	64.2857
42	99.4565	98.913	99.4565
43	85.9903	81.1594	67.1498
44	74.2515	65.2695	79.0419
45	55.2486	36.4641	28.7293
Avg	**67.466735**	**58.2341375**	**58.3991**

Table 69: User Individual 10-35 with 8 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
11	51.1521	41.9355	52.5346
12	70.202	35.3535	27.7778
13	90.4459	94.2675	78.3839
14	66.2857	69.7143	74.8571
15	39.5122	40	41.9512
16	69.6078	83.3333	63.7255
17	48.227	48.9362	52.4823
18	54.2484	48.366	52.2876
19	64.4737	69.7368	42.7632
20	82.6531	94.898	81.1224
21	40.9326	40.4145	54.4041
22	65.7459	75.1381	65.7459
23	41.7143	33.1429	49.1429
24	73.8636	73.2955	66.4773
25	56.875	42.5	56.875
26	86.9565	74.8792	65.2174
27	65.641	55.8974	47.6923
28	44.2748	64.1221	41.9847
29	77	65.9459	54.0541
30	82.2086	78.5276	76.0736
31	76.8421	60.5263	43.1579
32	67.1233	53.4247	74.6575
33	75.7062	58.1921	60.452
34	71.7791	81.5951	71.1656
35	95.679	95.679	85.1852
36	86.4322	80.402	76.8844
37	42.487	38.8601	55.9585
38	34.7826	51.6304	34.7826
39	62	64	55.5
40	42.2078	37.6623	37.6623
41	60	66.4286	58.5714
42	97.2826	99.4565	91.8478
43	82.6087	71.0145	71.0145
44	70.6587	54.491	72.4551
45	43.6464	44.7514	29.8343
Avg	**65.1872343**	**62.52909429**	**58.9909**

Table 70: User Individual 20-25 with 8 features

	BayesNet	**MultilayerPercepton**	**LMT**
Nr			
21	51.2953	46.114	43.0052
22	67.4033	78.453	62.9834
23	44.5714	52	45.1429
24	80.6818	80.6818	74.4318
25	58.125	59.375	66.875
26	87.9227	86.4734	61.8357
27	64.6154	65.641	35.8974
28	49.6183	67.9389	39.6947
29	87.027	79.4595	60.5405
30	76.0736	85.2761	71.7791
31	85.2632	97.8947	69.4737
32	67.1233	73.2877	61.6438
33	78.5311	92.0904	77.9661
34	74.8466	76.6871	66.2577
35	92.5926	93.8272	75.3086
36	89.9497	90.4523	81.9095
37	44.0415	49.7409	49.7409
38	38.0435	46.7391	37.5
39	64	60.5	52.5
40	37.6623	33.1169	29.2208
41	59.2857	83.5714	72.8571
42	98.913	97.8261	94.5652
43	73.913	92.2705	65.2174
44	64.6707	66.4671	67.0659
45	54.69611	56.3536	31.4917
Avg	**67.634644**	**72.489508**	**59.7962**

C Source Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;

namespace RenamingGazi
{
 class Program
 {
 static void Main(string[] args)
 {
 templatesIntoUserActivities();
 //foo();
 //personalthreetwelve();
 //globalSessionBased();
 //globalUserBased();
 //globalSessionIndividualBased();

  }



  static void foo()
  {
  List<string[]> norm = new List<string[]>();
  List<string[]> fast = new List<string[]>();
  List<string[]> slow = new List<string[]>();
   string folderFiles = @"E:\Test\features\";
   DirectoryInfo dir = new DirectoryInfo(folderFiles);
   FileInfo[] files = dir.GetFiles();

    foreach (FileInfo file in files)
    {
      string user = file.Name.Split('.')[0];
      using (TextReader tr = new StreamReader(file.FullName))
    {
     string line = "";
     while ((line = tr.ReadLine()) != null)
    {
     string[] info = line.Split('\t');
     if (info.Length == 5)
    {
     string activity = info[info.Length - 1];
     info[info.Length - 1] = user;
```

```
     if (activity.Equals("Norm"))
     norm.Add(info);
     else if (activity.Equals("Fast"))
     fast.Add(info);
     else if (activity.Equals("Slow"))
     slow.Add(info);

                         }

                     }

                 }
             }


 //Normal
using (TextWriter tw = new StreamWriter(folderFiles + "classificationByActivity\\
                 " + "norm.arff"))
 {
  createHeaderActivities(tw);

 foreach (string[] l in norm)
 {
  for (int i = 0; i < l.Length; i++)
   {
    if (i == l.Length - 1)
     tw.Write(l[i] + "\n");
    else
     tw.Write(l[i] + "\t");
                     }
                 }
             }

 //Slow
using (TextWriter tw = new StreamWriter(folderFiles + "classificationByActivity
        \\" + "slow.arff"))
  {
    createHeaderActivities(tw);

    foreach (string[] l in slow)
    {
     for (int i = 0; i < l.Length; i++)
      {
       if (i == l.Length - 1)
        tw.Write(l[i] + "\n");
        else
         tw.Write(l[i] + "\t");
                     }
                 }
             }

 //Fast
using (TextWriter tw = new StreamWriter(folderFiles + "classificationByActivity
```

```
        \\" + "fast.arff"))
 {
  createHeaderActivities(tw);

  foreach (string[] l in fast)
   {
    for (int i = 0; i < l.Length; i++)
     {
     if (i == l.Length - 1)
      tw.Write(l[i] + "\n");
     else
      tw.Write(l[i] + "\t");
                }
            }
        }





        }

/// <summary>
/// This method read all templates files and seperate them
///into each user with fast, normal, slow activities
/// </summary>

static void globalUserBased()
 {
string folderFiles = @"E:\Biometrics\Databases\Gait\
\Milestone-Gazmend-Activity110311\DataToAnalyse\walksRenamed\templates\";
 DirectoryInfo dir = new DirectoryInfo(folderFiles);
 FileInfo[] files = dir.GetFiles();
 List<List<string>> training = new List<List<string>>();
 List<List<string>> testing = new List<List<string>>();

 foreach (FileInfo file in files)
  {
  string user = file.Name.Split('_')[0];
  int userNumber = Convert.ToInt16(user);
  int session = Convert.ToInt16(file.Name.Split('_')[1]);

  if (userNumber <= 20)
   {
  List<List<string>> newList = new List<List<string>>();
  using (TextReader tr = new StreamReader(file.FullName))
   {
    string line = "";
    List<string> features = null;
    while ((line = tr.ReadLine()) != null)
   {
    features = new List<string>();
    string[] valuesString = line.Split(' ');
```

```
 double[] values = new double[valuesString.Length];

for (int i = 0; i < valuesString.Length; i++)
 {
  values[i] = Convert.ToDouble(valuesString[i]);
 }

 features.Add("" + stddev(values));
 features.Add("" + values.Min());
 features.Add("" + values.Max());
 features.Add("" + values.Length);
 features.Add("" + mean(values));
 features.Add("" + rootMeanSquare(values));
 features.Add("" + energy(values));
 features.Add("" + entropy(values));

 features.Add(file.Name.Split('_')[2]);
 training.Add(features);
 }

  }
   }
else
{

 List<List<string>> newList = new List<List<string>>();
 using (TextReader tr = new StreamReader(file.FullName))
{
 string line = "";
 List<string> features = null;
 while ((line = tr.ReadLine()) != null)
{
 features = new List<string>();
 string[] valuesString = line.Split(' ');
 double[] values = new double[valuesString.Length];

 for (int i = 0; i < valuesString.Length; i++)
{
 values[i] = Convert.ToDouble(valuesString[i]);
}

  features.Add("" + stddev(values));
  features.Add("" + values.Min());
  features.Add("" + values.Max());
  features.Add("" + values.Length);
  features.Add("" + mean(values));
  features.Add("" + rootMeanSquare(values));
  features.Add("" + energy(values));
  features.Add("" + entropy(values));

  features.Add(file.Name.Split('_')[2]);
          testing.Add(features);
}
```

```
      }


       }


        }

  int s = 0;

using (TextWriter tw = new StreamWriter(folderFiles +
       "features\\train.arff"))
  {
   createHeaderMultiple(tw);

   foreach (List<string> l in training)
   {

    for (int i = 0; i < l.Count; i++)
   {
    if (i == l.Count - 1)
     tw.Write(l[i] + "\n");
   else
     tw.Write(l[i] + "\t");
    }
      }


         }

using (TextWriter tw = new StreamWriter(folderFiles +
      "features\\test.arff"))
   {
    createHeaderMultiple(tw);

    foreach (List<string> l in testing)
   {
    for (int i = 0; i < l.Count; i++)
   {
    if (i == l.Count - 1)
     tw.Write(l[i] + "\n");
    else
     tw.Write(l[i] + "\t");
    }

     }
         }

         }

/// <summary>
```

```
/// This method read all templates files and seperate them
/// into each user with fast, normal, slow activities
/// </summary>

static void globalSessionBased()
 {
string folderFiles = @"E:\Biometrics\Databases\Gait\
\Milestone-Gazmend-Activity110311\DataToAnalyse\walksRenamed\templates\";
  DirectoryInfo dir = new DirectoryInfo(folderFiles);
  FileInfo[] files = dir.GetFiles();
  List<List<string>> training = new List<List<string>>();
  List<List<string>> testing = new List<List<string>>();

  foreach (FileInfo file in files)
   {
   string user = file.Name.Split('_')[0];
   int session = Convert.ToInt16(file.Name.Split('_')[1]);

   if (session <= 9)
    {
    List<List<string>> newList = new List<List<string>>();
    using (TextReader tr = new StreamReader(file.FullName))
    {
     string line = "";
     List<string> features = null;
     while ((line = tr.ReadLine()) != null)
    {
      features = new List<string>();
      string[] valuesString = line.Split(' ');
      double[] values = new double[valuesString.Length];

     for (int i = 0; i < valuesString.Length; i++)
      {
       values[i] = Convert.ToDouble(valuesString[i]);
      }

       features.Add("" + stddev(values));
       features.Add("" + values.Min());
       features.Add("" + values.Max());
       features.Add("" + values.Length);
       features.Add("" + mean(values));
       features.Add("" + rootMeanSquare(values));
       features.Add("" + energy(values));
       features.Add("" + entropy(values));

       features.Add(file.Name.Split('_')[2]);
           training.Add(features);
     }
       }

         }
      else
     {
```

```
      List<List<string>> newList = new List<List<string>>();
      using (TextReader tr = new StreamReader(file.FullName))
      {
       string line = "";
       List<string> features = null;
       while ((line = tr.ReadLine()) != null)
      {
       features = new List<string>();
       string[] valuesString = line.Split(' ');
       double[] values = new double[valuesString.Length];

       for (int i = 0; i < valuesString.Length; i++)
      {
       values[i] = Convert.ToDouble(valuesString[i]);
     }

      features.Add("" + stddev(values));
      features.Add("" + values.Min());
      features.Add("" + values.Max());
      features.Add("" + values.Length);
      features.Add("" + mean(values));
      features.Add("" + rootMeanSquare(values));
      features.Add("" + energy(values));
      features.Add("" + entropy(values));

      features.Add(file.Name.Split('_')[2]);
        testing.Add(features);
       }


        }


          }


            }

   int s = 0;

using (TextWriter tw = new StreamWriter(folderFiles +
      "features\\train.arff"))
  {
    createHeaderMultiple(tw);


  foreach (List<string> l in training)
   {

    for (int i = 0; i < l.Count; i++)
     {
      if (i == l.Count - 1)
```

```
        tw.Write(l[i] + "\n");
      else
       tw.Write(l[i] + "\t");
      }
        }


        }

using (TextWriter tw = new StreamWriter(folderFiles +
       "features\\test.arff"))
   {
    createHeaderMultiple(tw);

    foreach (List<string> l in testing)
   {
    for (int i = 0; i < l.Count; i++)
   {
    if (i == l.Count - 1)
     tw.Write(l[i] + "\n");
    else
     tw.Write(l[i] + "\t");
    }

      }
        }

         }
/// <summary>
/// This method read all templates files and seperate them
///into each user with fast, normal, slow activities
/// </summary>

static void globalSessionIndividualBased()
{
string folderFiles = @"E:\Biometrics\Databases\Gait\
\Milestone-Gazmend-Activity110311\DataToAnalyse\walksRenamed\templates\";
  DirectoryInfo dir = new DirectoryInfo(folderFiles);
  FileInfo[] files = dir.GetFiles();
  List<List<string>> training = new List<List<string>>();
  Dictionary<string, List<List<string>>> testing = new Dictionary<string,
        List<List<string>>>();

  foreach (FileInfo file in files)
  {
   string user = file.Name.Split('_')[0];
   int userNumber = Convert.ToInt16(user);
   int session = Convert.ToInt16(file.Name.Split('_')[1]);

// Session or user Number
   //if(session <= 6)
   if (userNumber <= 20)
   {
```

```
    List<List<string>> newList = new List<List<string>>();
    using (TextReader tr = new StreamReader(file.FullName))
  {
   string line = "";
   List<string> features = null;
   while ((line = tr.ReadLine()) != null)
  {
   features = new List<string>();
   string[] valuesString = line.Split(' ');
   double[] values = new double[valuesString.Length];

  for (int i = 0; i < valuesString.Length; i++)
   {
    values[i] = Convert.ToDouble(valuesString[i]);
   }

    features.Add("" + stddev(values));
    features.Add("" + values.Min());
    features.Add("" + values.Max());
    features.Add("" + values.Length);
    features.Add("" + mean(values));
    features.Add("" + rootMeanSquare(values));
    features.Add("" + energy(values));
    features.Add("" + entropy(values));

   features.Add(file.Name.Split('_')[2]);
   training.Add(features);
    }

     }

      }
       else
       {

       if (!testing.ContainsKey(user))
       {
        List<List<string>> newList = new List<List<string>>();
        using (TextReader tr = new StreamReader(file.FullName))
       {
        string line = "";
        while ((line = tr.ReadLine()) != null)
       {
        List<string> features = new List<string>();
        string[] valuesString = line.Split(' ');
        double[] values = new double[valuesString.Length];

        for (int i = 0; i < valuesString.Length; i++)
         {
           values[i] = Convert.ToDouble(valuesString[i]);
         }
          features.Add("" + stddev(values));
          features.Add("" + values.Min());
```

```
            features.Add("" + values.Max());
            features.Add("" + values.Length);
            features.Add("" + mean(values));
            features.Add("" + rootMeanSquare(values));
            features.Add("" + energy(values));
            features.Add("" + entropy(values));

          features.Add(file.Name.Split('_')[2]);
                  newList.Add(features);

        }
            testing.Add(user, newList);

      }
          }
          else
           {

        List<List<string>> alreadyList = testing[user];
        using (TextReader tr = new StreamReader(file.FullName))
         {
          string line = "";
          while ((line = tr.ReadLine()) != null)
         {
          List<string> features = new List<string>();
          string[] valuesString = line.Split(' ');
          double[] values = new double[valuesString.Length];

          for (int i = 0; i < valuesString.Length; i++)
         {
          values[i] = Convert.ToDouble(valuesString[i]);
         }
          features.Add("" + stddev(values));
          features.Add("" + values.Min());
          features.Add("" + values.Max());
          features.Add("" + values.Length);
          features.Add("" + mean(values));
          features.Add("" + rootMeanSquare(values));
          features.Add("" + energy(values));
          features.Add("" + entropy(values));

          features.Add(file.Name.Split('_')[2]);
                   alreadyList.Add(features);


           }
           testing.Remove(user);
          testing.Add(user, alreadyList);
           }
               }

                 }
             }
```

```
  int s = 0;

using (TextWriter tw = new StreamWriter(folderFiles +
       "features\\train.arff"))
 {
  createHeaderMultiple(tw);

  foreach (List<string> l in training)
  {
   for (int i = 0; i < l.Count; i++)
  {
   if (i == l.Count - 1)
    tw.Write(l[i] + "\n");
  else
   tw.Write(l[i] + "\t");
  }
    }


     }

  foreach (KeyValuePair<string, List<List<string>>> pair in testing)
   {

    List<List<string>> info = pair.Value;

 using (TextWriter tw = new StreamWriter(folderFiles + "features\\" +
        pair.Key + "_test.arff"))
   {
    createHeaderMultiple(tw);

    foreach (List<string> l in info)
   {
    for (int i = 0; i < l.Count; i++)
   {
    if (i == l.Count - 1)
      tw.Write(l[i] + "\n");
    else
      tw.Write(l[i] + "\t");
     }
       }
        }

           }


        }

/// <summary>
/// This method read all templates files and seperate them into
///each user with fast, normal, slow activities
/// </summary>
```

```
static void personalthreetwelve()
 {
 string folderFiles = @"E:\Biometrics\Databases\Gait\
\Milestone—Gazmend—Activity110311\DataToAnalyse\walksRenamed\templates\";
 DirectoryInfo dir = new DirectoryInfo(folderFiles);
 FileInfo[] files = dir.GetFiles();
 Dictionary<string, List<List<string>>> training = new Dictionary<string,
 List<List<string>>>();
 Dictionary<string, List<List<string>>> testing = new Dictionary<string,
 List<List<string>>>();

 foreach (FileInfo file in files)
  {
   string user = file.Name.Split('_')[0];
   int session = Convert.ToInt16(file.Name.Split('_')[1]);
   if (session <= 9)
  {
   if (!training.ContainsKey(user))
  {
   List<List<string>> newList = new List<List<string>>();
   using (TextReader tr = new StreamReader(file.FullName))
  {
    string line = "";
    while ((line = tr.ReadLine()) != null)
  {
    List<string> features = new List<string>();
    string[] valuesString = line.Split(' ');
    double[] values = new double[valuesString.Length];

   for (int i = 0; i < valuesString.Length; i++)
    {
     values[i] = Convert.ToDouble(valuesString[i]);
    }

     features.Add("" + stddev(values));
     features.Add("" + values.Min());
     features.Add("" + values.Max());
     features.Add("" + values.Length);
     features.Add(file.Name.Split('_')[2]);
     newList.Add(features);


    }
     training.Add(user, newList);
    }

      }
     else
     {


   List<List<string>> alreadyList = training[user];
```

```
using (TextReader tr = new StreamReader(file.FullName))
{
  string line = "";
  while ((line = tr.ReadLine()) != null)
{
 List<string> features = new List<string>();
 string[] valuesString = line.Split(' ');
 double[] values = new double[valuesString.Length];

 for (int i = 0; i < valuesString.Length; i++)
 {
 values[i] = Convert.ToDouble(valuesString[i]);
  }

  features.Add("" + stddev(values));
  features.Add("" + values.Min());
  features.Add("" + values.Max());
  features.Add("" + values.Length);
  features.Add(file.Name.Split('_')[2]);
  alreadyList.Add(features);


   }

   training.Remove(user);
   training.Add(user, alreadyList);
                    }

                }

             }
else
{
if (!testing.ContainsKey(user))
{
 List<List<string>> newList = new List<List<string>>();
 using (TextReader tr = new StreamReader(file.FullName))
{
 string line = "";
 while ((line = tr.ReadLine()) != null)
{
 List<string> features = new List<string>();
 string[] valuesString = line.Split(' ');
 double[] values = new double[valuesString.Length];

 for (int i = 0; i < valuesString.Length; i++)
  {
  values[i] = Convert.ToDouble(valuesString[i]);
  }
   features.Add("" + stddev(values));
   features.Add("" + values.Min());
   features.Add("" + values.Max());
   features.Add("" + values.Length);
```

```
  features.Add(file.Name.Split('_')[2]);
       newList.Add(features);


 }
   testing.Add(user, newList);

 }
   }
 else
{


 List<List<string>> alreadyList = testing[user];

using (TextReader tr = new StreamReader(file.FullName))
 {
  string line = "";
  while ((line = tr.ReadLine()) != null)
 {
 List<string> features = new List<string>();
 string[] valuesString = line.Split(' ');
 double[] values = new double[valuesString.Length];

for (int i = 0; i < valuesString.Length; i++)
{
 values[i] = Convert.ToDouble(valuesString[i]);
}

features.Add("" + stddev(values));
features.Add("" + values.Min());
features.Add("" + values.Max());
features.Add("" + values.Length);
features.Add(file.Name.Split('_')[2]);
 alreadyList.Add(features);


 }

 testing.Remove(user);
 testing.Add(user, alreadyList);

 }

   }
     }


       }

int s = 0;

foreach (KeyValuePair<string, List<List<string>>> pair in training)
```

```
    {

     List<List<string>> info = pair.Value;

 using (TextWriter tw = new StreamWriter(folderFiles + "features\\" +
       pair.Key + "_train.arff"))
      {
        createHeader(tw);

        foreach (List<string> l in info)
        {
          for (int i = 0; i < l.Count; i++)
        {
          if (i == l.Count - 1)
          tw.Write(l[i] + "\n");
          else
          tw.Write(l[i] + "\t");
                      }
                  }
              }


          }

    foreach (KeyValuePair<string, List<List<string>>> pair in testing)
    {

    list<List<string>> info = pair.Value;
 using (TextWriter tw = new StreamWriter(folderFiles + "features\\" +
       pair.Key + "_test.arff"))
     {
      createHeader(tw);

      foreach (List<string> l in info)
       {
        for (int i = 0; i < l.Count; i++)
        {
         if (i == l.Count - 1)
         tw.Write(l[i] + "\n");
         else
         tw.Write(l[i] + "\t");
                      }
                  }
              }


          }
       }

/// <summary>
/// This method read all templates files and seperate them
///into each user with fast, normal, slow activities
/// </summary>
```

```
static void templatesIntoUserActivities()
{
string folderFiles = @"E:\Biometrics\Databases\Gait\
\Milestone—Gazmend—Activity110311\DataToAnalyse\walksRenamed\templates\";
 DirectoryInfo dir = new DirectoryInfo(folderFiles);
 FileInfo[] files = dir.GetFiles();
 Dictionary<string, List<List<string>>> information = new Dictionary<string,
 List<List<string>>>();

 foreach (FileInfo file in files)
 {
  string user = file.Name.Split('_')[0];

  if (!information.ContainsKey(user))
   {
    List<List<string>> newList = new List<List<string>>();
    using (TextReader tr = new StreamReader(file.FullName))
   {
    string line = "";
    while ((line = tr.ReadLine()) != null)
   {
    List<string> features = new List<string>();
    string[] valuesString = line.Split(' ');
    double[] values = new double[valuesString.Length];

    for (int i = 0; i < valuesString.Length; i++)
     {
    values[i] = Convert.ToDouble(valuesString[i]);
     }


    features.Add("" + stddev(values));
    features.Add("" + values.Min());
    features.Add("" + values.Max());
    features.Add("" + values.Length);
    features.Add("" + mean(values));
    features.Add("" + rootMeanSquare(values));
   //features.Add("" + energy(values));
   //features.Add("" + entropy(values));

    features.Add(file.Name.Split('_')[2]);
      newList.Add(features);


    }

    information.Add(user, newList);

     }

       }
   else
     {
```

```
List<List<string>> alreadyList = information[user];

using (TextReader tr = new StreamReader(file.FullName))
 {
  string line = "";
  while ((line = tr.ReadLine()) != null)
 {
 List<string> features = new List<string>();
string[] valuesString = line.Split(' ');
double[] values = new double[valuesString.Length];

for (int i = 0; i < valuesString.Length; i++)
{
 values[i] = Convert.ToDouble(valuesString[i]);
}
 features.Add("" + stddev(values));
 features.Add("" + values.Min());
 features.Add("" + values.Max());
 features.Add("" + values.Length);
 features.Add("" + mean(values));
 features.Add("" + rootMeanSquare(values));
//features.Add("" + energy(values));
//features.Add("" + entropy(values));

features.Add(file.Name.Split('_')[2]);
  alreadyList.Add(features);


 }


 information.Remove(user);
 information.Add(user, alreadyList);

}

  }


    }

foreach (KeyValuePair<string, List<List<string>>> pair in information)
 {

  List<List<string>> info = pair.Value;

using (TextWriter tw = new StreamWriter(folderFiles + "features\\" +
      pair.Key + ".arff"))
  {
   //createHeader(tw);
   createHeaderMultiple(tw);

   foreach (List<string> l in info)
```

```
{
 for (int i = 0; i < l.Count; i++)
{
 if (i == l.Count - 1)
 tw.Write(l[i] + "\n");
 else
 tw.Write(l[i] + "\t");
                  }
              }
          }


      }
  }

static void createHeaderMultiple(TextWriter tw)
{
 tw.WriteLine("@relation features");
 tw.WriteLine("");
 tw.WriteLine("@attribute stddev real");
 tw.WriteLine("@attribute min real");
 tw.WriteLine("@attribute max real");
 tw.WriteLine("@attribute clength real");
 tw.WriteLine("@attribute mean real");
 tw.WriteLine("@attribute rms real");
//tw.WriteLine("@attribute energy real");
//tw.WriteLine("@attribute entropy real");
  tw.WriteLine(""); tw.WriteLine("");
  tw.WriteLine("@attribute class {Norm,Fast,Slow}");
  tw.WriteLine("");
  tw.WriteLine("@data");
  tw.WriteLine("");

    }

static void createHeader(TextWriter tw)
 {
 tw.WriteLine("@relation features");
 tw.WriteLine("");
 tw.WriteLine("@attribute stddev real");
 tw.WriteLine("@attribute min real");
 tw.WriteLine("@attribute max real");
 tw.WriteLine("@attribute clength real");
 tw.WriteLine(""); tw.WriteLine("");
 tw.WriteLine("@attribute class {Norm,Fast,Slow}");
 tw.WriteLine("");
 tw.WriteLine("@data");
 tw.WriteLine("");

    }

static void createHeaderActivities(TextWriter tw)
 {
```

```
tw.WriteLine("@relation features");
tw.WriteLine("");
tw.WriteLine("@attribute stddev real");
tw.WriteLine("@attribute min real");
tw.WriteLine("@attribute max real");

      tw.WriteLine("@attribute clength real");
tw.WriteLine(""); tw.WriteLine("");
tw.WriteLine("@attribute class {");
 for (int i = 1; i <= 45; i++)
  {
    if (i == 45)
     tw.Write(i + "}");
    else if (i < 10)
     tw.Write("0" + i + ",");
    else
     tw.Write(i + ",");
   }
   tw.WriteLine("");
   tw.WriteLine("@data");
   tw.WriteLine("");
  }

static void createHeaderActivitiesEar(TextWriter tw)
{
 tw.WriteLine("@relation features");
 tw.WriteLine("");
 tw.WriteLine("@attribute stddev real");
 tw.WriteLine("@attribute min real");
 tw.WriteLine("@attribute max real");
 tw.WriteLine(""); tw.WriteLine("");
 tw.WriteLine("@attribute class {");
 for (int i = 1; i <= 50; i++)
  {
   if (i == 50)
   tw.Write(i + "}");
   else if (i < 10)
   tw.Write("0" + i + ",");
   else
   tw.Write(i + ",");
  }
  tw.WriteLine("");
  tw.WriteLine("@data");
  tw.WriteLine("");
  }


static double stddev(double[] vector)
 {
  double sum = 0.0, sumOfSqrs = 0.0;
  for (int i = 0; i < vector.Length; i++)
 {
  sum += Convert.ToDouble(vector[i]);
```

```
      sumOfSqrs += Math.Pow(Convert.ToDouble(vector[i]), 2.0);
      }
      double topSum = (vector.Length * sumOfSqrs) - (Math.Pow(sum, 2));
      double n = (double)vector.Length;
      return Math.Sqrt(topSum / (n * (n - 1)));
      }

     public static double mean(double[] vector)
     {
      double mean = 0;
      foreach (double average in vector)
     {
      mean += average;
     }

      mean /= vector.Length;
     return mean;
         }

private static double rootMeanSquare(double[] vector)
 {
  double sum = 0;
  for (int i = 0; i < vector.Length; i++)
 {
   sum += (vector[i] * vector[i]);
 }
   return Math.Sqrt(sum / vector.Length);
 }

  private static double energy(double[] vector)
{
  double energy = 0;

  for (int i = 0; i < vector.Length; i++)
  {
  energy += (vector[i] * vector[i]);
 }

   return energy;
 }

  private static double entropy(double[] vector)
   {
     double entropy = 0;
     double X = 0;

     for (int i = 0; i < vector.Length; i++)
   {
     X += Math.Abs(vector[i]);
   }

   for (int i = 0; i < vector.Length; i++)
    {
```

```
        double pi = vector[i] / X;
        entropy += pi * Math.Log(pi);
      }

        return entropy;
      }


    }
  }
```

D Accepted Paper

Towards an Automatic Gait Recognition System using Activity Recognition (Wearable Based)

Gazmend Bajrami
Norwegian Information Security Lab.
Gjøvik University College, Norway
Email: gazmend.bajrami@hig.no

Mohammad Omar Derawi
Norwegian Information Security Lab.
Gjøvik University College, Norway
Email: mohammadd.derawi@hig.no

Patrick Bours
Norwegian Information Security Lab.
Gjøvik University College, Norway
Email: patrick.bours@hig.no

***Abstract*—The need of increasing the security measures in mobile devices has led researchers focus on finding new security mechanisms. In this paper we propose a solution to keep data secured by ensuring that only the authorized user can access the data in a mobile phone. By using gait recognition as an important element for the authentication process we propose an automatic gait recognition system to be used for continuous authentication. Since recent gait recognition research only focuses on manual extraction of walking activities from the accelerometer signal, a solution to this issue could be activity recognition that would reduce the disadvantages of gait recognition by identifying the activities of a person continuously and automatically. Activity recognition would not only make it possible to authenticate the user in different daily activities like slow walking, normal walking, fast walking even running, but also help in avoiding authentication when the user is in passive state like sitting, standing still, etc.. This is one of the key factors and an interesting challenge which would benefit the data security area.**

***Index Terms*—activity recognition, gait recognition, mobile devices, accelerometers**

I. Introduction

In recent years, activity recognition has become a very important field of research due to its application in many different areas such as health care, fitness, industrial application, security, entertainment, etc.. The goal of activity recognition is to recognize and track human activities, which is also an important goal of ubiquities computing [1]. Computers are becoming more pervasive in modern society by integrating in our phones, music players, cars etc.. The idea of ubiquitous computing is to integrate computers into our environment, everyday objects and activities etc., to become assistance in our everyday lives and work [2].

Today, whenever we use computer systems, they demand authentication as a measure of security. Typically, we perform the authentication at login time with either a password, token, biometric characteristic and/or a combination of these. Performing the last mentioned measure is a stronger guarantee that the claimed user logging in is not an impostor but an authorized user. An issue raises that not many systems of security requires any further measures once the user is granted access (thus assuming that the user is continuously legitimated into the system). Continuous authentication insurance of the user's legitimacy is of high importance in critical or high security environments, this means that it is necessary to continuously ensure that the user is the legitimated one. Therefore, performing the user authentication continuously while the system is actively used is something essential. Nevertheless, this kind of authentication needs to be "attractive" for the user. A very good solution for continues authentication is activity recognition from the gait signal.

The latest generation of mobile devices (smart phones, PDA etc.) are more sophisticated and they come with built-in sensors like accelerometers, gyro-scopes, Global Positioning Systems (GPS), etc., for example accelerometers can record the motion of the body and provide sufficient data that can be used for recognizing activities. Thus, accelerometers are becoming a very important tool due to good results in activity recognition: they are cheap, small, effective, require little energy, they are not sensitive on the conditions of the environment etc.. Because of these advantages of accelerometers they are incorporated in newer mobile devices, e.g. the iPhones, iPod, iPad, HTC etc..

This paper is divided into three sections. Section II gives an elaborate study related work overview on activity recognition to be used for gait recognition. Although gait recognition is not mentioned in this paper, an extensive survey of gait recognition can be found in [3]. Section III proposes a solution for a full automatic wearable based gait recognition system. Finally, Section IV concludes the paper.

II. Activity Recognition - Related Work

Activity recognition is the process of identifying everyday common human activities in real life. It is a new area of study, and is becoming an interesting research field due to different areas of application. Accelerometers come integrated on new models of mobile devices such as smart phones, tablet computers, digital audio players (Ipod) etc., which record the body motion. The majority of studies for activity recognition are performed by using wearable sensors. Several studies have shown that wearable sensors are adequate for activity recognition. In the following we will show some of the sensors that have been used so far for activity recognition, a summary of different activities that were recognized by using various sensors and the approaches used for identifying different human activities.

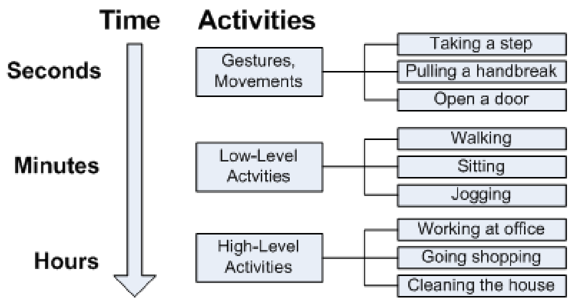

Fig. 1. Level of Activities [2]

Due to many different application areas of activity recognition, there is no surprise that the list of activities that many researchers have tried to recognize with various sensors is long. According to [2], activities can be categorized in three groups based on duration and/or complexity: *Gestures (or Movement/Motif)*, *Low-Level Activities*, and *High-Level Activities*. Activities such as walking, sitting, standing, eating, cleaning windows are considered as low-level activities which usually last between seconds and several minutes. As high-level activities are considered activities like sightseeing, cleaning the house, working at office, that usually last for more than a few minutes up to a few hours. Figure 1 illustrates these groups of activities.

A. Experiments (Activities)

The identification of everyday routine and leisure activities such as walking, running, biking, sitting, climbing and lying have already been analyzed in laboratory settings by several researchers. All these studies were done by different sensors such as accelerometers which were embedded in wearable sensing devices to collect the needed data. The types of sensors used for activity recognition are to be discussed in the next section. Accelerometer sensors are very useful for low-powered equipments like smart phones, tablet computers with applications that are suitable for real-time detection of user's activities. Physical activities such as walking, walking up/down stairs standing, sitting, and running have been studied by some of the researchers using different accelerometers sensors. Table I summaries different activities by different studies.

Another class of activities, mainly studied in healthcare environments, are the so-called "Activities of Daily Living" (ADLs). ADLs include activities such like bathing, toileting, dressing, feeding ourselves, homemaking which are basic skills needed for daily self-care activities. A set of ADLs is known as the "Instrumental Activities of Daily Living" (IADLs), those are skills beyond basic self-care which a person needs to perform for an independent living. IADLs include activities like shopping, driving, cleaning, cooking, doing laundry and managing money. Table II shows an overview of these activities.

TABLE I
ACTIVITY RECOGNITION RESEARCH STUDIES. #TP = TEST PERSONS

Study	Activities	#TP
[4]	walking flat, walking slope-up, slope-down, walking stairs	52
[5]	sitting, walking, jogging, walking stairs, standing	29
[6]	sitting, standing, and walking	26
[7]	walking, running, cycling	24
[8]	walking, running, sitting, standing, bicycling	20
[9]	walking, climbing stairs	15
[10]	lying down, sitting and standing, walking, running,	12
[11]	sitting, standing, walking, walking stairs, riding elevator up/down, and brushing teeth	12
[1]	running, still, jumping and walking	11
[12]	sitting, walking, walking (street), waiting at a tram stop, riding a tram	8
[13]	walking, standing, sitting and running, walking stairs	6
[14]	sitting, walking, running, walking stairs	6
[15]	standing, walking, running, climbing	5
[16]	standing, sitting, lying, walking, running	5
[17]	sitting, walking, jogging, riding a bike, walking stairs	2

TABLE II
STUDIES OF ACTIVITY RECOGNITION OF DAILY LIVING (ADL)

Study	Activities (ADL)	#TP
[18]	toileting, washing, housework, leisure activity, oral hygiene, heating use, taking medication, etc.	14
[19]	mopping, cleaning windows, making bed, watering plants, washing dishes, setting the table, vacuuming, ironing, dusting	12
[10]	lying, rowing, cycling (training,regular), sitting, standing, running, walking, football	12
[20]	prepare food, clean dishes, wash clothes	10
[21]	showering, urination, flushing, washing Hands, defecation, brushing teeth	4
[22]	prepare food, toileting, bathing, dressing, grooming, preparing a beverage, doing laundry, etc.	2
[23]	prepare different food, eat cereal, dust, brush teeth, tend plants, set table, clean windows, take medication, shower, shave	2

B. Data Acquisition

Depending on the activities there have been used several kinds of sensors in the data acquisition process for activity recognition. As mentioned earlier, accelerometer sensors are adequate and most commonly used for continues activity recognition. They are also considered to be less intrusive than other sensors such as RFID gloves, microphones, and cameras [2]. Therefore, accelerometers are becoming very important tools due to many advantages in activity recognition. There is not a single sensor that can record all the body movements and recognize all kind of human everyday activities at one time. Therefore, most researches today have been using different sensors to capture the data and multiple sensors attached on multiple parts of the body such as, hip, wrist, arm, ankle, chest, thigh, knee. For instance, activities like walking fast, walking slow, and running can be recognized by motion sensors but these sensors can not recognize activities such as, talking, reading, driving car etc.. Table III overviews some of the most widely used sensors for activity recognition research.

Other sensors that have been used for activity recognition

TABLE III
SENSORS USED IN DIFFERENT STUDIES.

Study	Sensor Placement	Sensor
[24]	Above ankle, above knee,hip, wrist,elbow,	3D Accelerometer (ADXL311)
[25]	Belt (left/right)	3D Accelerometer ADXL202
[26]	Chest	3D Accelerometers (ADXL213, analog)
[8]	Hip, thigh, ankle, arm, wrist	2D Accelerometer (ADXL210E, analog)
[27]	Legs	2D accelerometer (ADXL202JE, analog) and Ball Switches
[28]	Legs (upper), above knee	1D Accelerometer (ADXL05s, analog) , passive infrared sensors, carbon monoxide sensor, microphones, pressure sensors, temperature sensors, touch-sensors and light-sensors
[29]	Near pelvic region	3D Accelerometer (CDXL04M3)
[1]	Pocket	3D Accelerometer (ADXL330, analog)
[5]	Pocket	3D Accelerometer (Cell phone)
[30]	Pocket	2D Accelerometer (ADXL202), GPS
[31]	Shoulder	Sociometer (IR transceiver, a microphone, two accelerometers, on-board storage, and power supply)
[32]	Waist	3D Accelerometer
[33]	Waist	3D Accelerometer and a microphone.
[34]	Waist belt	3D Accelerometer
[35]	Wrist, hip and thigh	2D accelerometer (ADXL202JE), Tilt switches

are: GPS sensors [10], vision sensors (i.e., cameras) [10], [36], microphones [21], [37], RFID tag readers [38], [18], [19], ball switches [27], fibber optical sensors [39], gyroscope [40], body and skin temperature sensors [28], [41], [42], [43], [10], light sensors [28], [41], [13], [44], foam pressure sensors [45], pressure sensors [41], physiological sensors [46], humidity and barometric sensors [41].

C. Activity Recognition Process

1) Segmentation: Detection of activities from the collected data is the process of finding the ”boundaries” for different activities in the accelerometer signal. Segmentation is a necessary step in the data analysis process before the feature extraction and the classification. Several segmentation techniques have been used to identify different activities from the sensor data. Some of the segmentation methods that have been used for activity recognition are: ”Sliding Windows”, ”Top-Down”, ”Bottom-Up” and ”Sliding Window and Bottom-Up (SWAB)” [47].

2) Feature Extraction: The input data recorded with the sensors from the human body motions is too large for processing, thus it is easier as an initial step to transform the large input data into a reduced representation set of features before further processing. The process of transforming the large input data into the set of features is called feature extraction. The feature extraction is a very important step; therefore features should be carefully chosen in order to extract relevant information from the input data, because it will have a strong influence in the results of classification. Features selection is an important and essential step in the design of any activity recognition system, in order to design an effective system. The features in different studies were analyzed mainly in time-domain and frequency-domain. In the following we will brief describe features extraction in the time-domain and frequency-domain.

Feature extraction in the Time-Domain: In much of the research, studies were considering only time-domain features due to avoid the complexity of pre-processing that required transformation of the signal into frequencies. They consume little processing power and the algorithms can be applied directly. Table IV shows a summary of papers that consider the time domain features.

TABLE IV
FEATURE EXTRACTION STUDIES IN THE TIME DOMAIN

Study	Approaches
[48], [8], [49], [24], [29], [50], [51], [52], [13], [53], [16], [54], [55], [23], [19], [56], [17]	Mean
[24], [54], [29], [48], [33], [53], [50], [51], [52], [13], [23], [19], [56], [17]	Variance or standard deviation
[51], [13], [50], [55]	Root mean square (RMS)
[55], [13], [48], [53], [44]	Zero or Mean Crossing Rate
[53], [51], [28], [55]	Derivative
[27], [57], [58], [59], [60]	Peak Count and Amplitude

Feature extraction in the Frequency-Domain: Unlike the time-domain features, the signal should transform data into the frequency domain and this process requires pre-processing and different transformations such as the use of Fast Fourier transform (FFT). Table V shows the most widely used features in frequency domain.

TABLE V
FEATURE EXTRACTION STUDIES IN THE FREQUENCY DOMAIN

Study	Approaches
[42], [28], [23], [19], [56], [37], [17], [52], [11], [61]	Fast Fourier Transform
[23], [19], [56], [29], [52], [11]	Energy
[23], [19], [56], [17], [62], [52], [11]	Spectral Entropy
[40], [62]	Frequency range power

3) Classification: Next step after the feature extraction is the classification process. In the classification process, the classification algorithm builds up a model (classifiers) for different human activities and then uses these classifier to identify human activities from the test data. A wide range of machine learning approaches and algorithms are used for activity recognition. Most of these approaches have been used for activity recognition which can be categorized into two groups: supervised learning and unsupervised learning.

Supervised learning is a machine learning technique, also sometimes called ”learning with a teacher” in which the system is trained by using a set of training data before it comes into use in classifying the test data. There are two

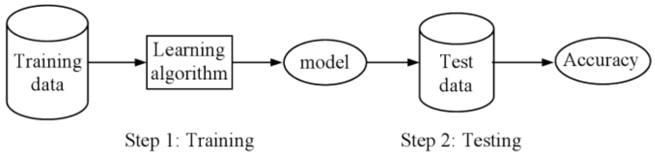

Fig. 2. The basic of learning process: training and testing [63]

general phases in a supervised learning technique: training and testing. During the training phase the system is taught (trained) by using a set of training data to create a classification model to classify unknown data. During the testing phase, the model of the system is tested using a set of test data to measure the classification accuracy [63]. Training and testing phases are illustrated in Figure 2.

The majority of works in activity recognition have been done by using supervised learning methods. A summary of these approaches applied so far is shown in Table VI. Supervised learning techniques are mostly used for activity recognition in majority of the researches. Next step is to look at the unsupervised learning techniques which are dissimilar than the supervised learning.

TABLE VI
SUPERVISED LEARNING APPROACHES USED FOR ACTIVITY RECOGNITION

Study	Approaches
[60], [8], [64], [29]	Naive Bayes Classifier
[8], [13], [29]	C4.5 Decision Tree
[40], [35], [28], [29]	Nearest Neighbor
[59], [17], [38]	Hidden Markov Model
[35], [29]	Support Vector Machine
[28]	Kohonen Self-Organising Map

Unsupervised learning by contrast does not use any training or testing data. Instead, it "tries" to classify the unknown data by separating the data into different classes (clusters). It is a "learning without teacher" method. The method tries to directly build models not basing itself on any priori-built model or knowledge. It learns from the unlabeled data, the task of this method is to discover classes of similar examples from the unlabeled data and organizes data into similarity groups, which is known as clustering, or by estimating the distribution of data within the input space which is called density estimation [65]. Clustering is the process of organizing unlabeled data into clusters, where the data in the same cluster are similar to each other and the data in different clusters are dissimilar [66]. A summary of the unsupervised learning approaches that are applied for activity recognition is shown in Table VII.

The process flow for unsupervised learning is illustrated in Figure 3.

D. Activity Recognition Performances

Studies have shown different accuracies for activity recognition systems in which the data collection was performed in a controlled laboratory settings (subjects are told how to walk,

TABLE VII
UNUPERVISED LEARNING APPROACHES USED FOR ACTIVITY RECOGNITION

Study	Approaches
[67], [68], [69]	Hidden Markov Model (HMM)
[70]	Hierarchies of HMM
[71]	Hierarchical Dynamic Bayesian Network
[37]	Multiple Eigenspaces
[69]	Gaussian Mixture Models
[72]	Multi-layered FSM

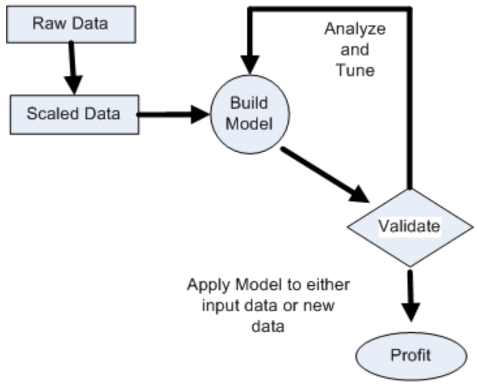

Fig. 3. Unsupervised Learning Workflow [73]

run etc.), from the experiments in which the data was collected under normal circumstances. As we saw in the data collection section a range of different sensors are used to collect the data. Experiments were performed by placing these sensors in one or multiple locations on the body. A summary of recognition accuracies is shown in the Table VIII.

TABLE VIII
RECOGNITION ACCURACIES.

Study	Recognition Accuracy	Activities Recognized	#TP
[7]	80%	walking, running, cycling, driving, sports	24
[8]	84%	walking, sitting, standing, running, computer work, bicycling, Lying down, etc.	20
[25]	83% - 90%	walking, downstairs, upstairs, opening doors	6
[74]	90%	walking, jogging, upstairs, downstairs, sitting, standing	29
[75]	90.8%	walking (slow, normal, fast), sitting, standing, lying, falling	6
[58]	92.85% - 95.91%	sitting, standing, walking,	8
[24]	65% - 95%	sitting, standing, walking, stairs up/down, whiteboard writing, shake hands, keyboard typing	1
[1]	97,51%	walking, jumping, still, running	11
[16]	99,5%	standing, sitting, lying, walking, running	5

III. Scenario and Proposal

A full automatic wearable sensor (WS) based gait recognition system using mobile devices is proposed in this section. The WS based recognition approach is the newest approach among the gait recognition methods available, i.e Machine Vision (MV) Based and Floor Sensor (FS) Based. WS is based on wearing motion recording sensors on the body of the person in different places; on the waist, pockets, shoes and so forth. Since wearable gait recognition system do not offer a full automatic mechanism today we will in this section give a possible solution to how this system is to be designed and implemented by including activity recognition as a major main step. Sensor based biometric gait research shows an increase in performance over time since 2005 where external dedicated sensors were applied until today where mobile phone accelerometers are being used. And to make gait recognition more stable, some issues need to be taken into consideration which we will see at the following subsections.

A. Scenario

We will here give some examples on different scenarios where activity recognition and gait recognition would make phones applicable as a security mechanism.

- **Shopping:** When a person is shopping, he or she is performing a lot of walking and standing. Since the user is constantly watching out for new equipments or clothing it will simply mean that the person is performing different activities by walking from one shop to another, from one cashier to another, etc.. In this case we can protect data of the person to ensure security of the phone.
- **Going to Work** People go to work by different means of transport. Some people use car, bicycle or even their motorbike. Since the mobile phone might be lost while walking out of the car or bicycling, it can ensure security. However, if a person is sitting in the car and the phone is standing still, the phone will also recognize that a ”standing still” activity is ongoing, and thus the phone should not be used at all for authentication. In this case, a backup solution should be applied such as using the PIN-code.
- **Fitness/Jogging** Even when people are making fitness, they might loose their phone when running outside their home. Running is still an activity and can also be used as a security mechanism towards authentication to the phone for usage.

These examples are only few out of many. An illustration of which activities can be recognized from gait signal data is shown in Figure 4. The interesting point of view here is that the mobile phone by using activity recognition for identifying activities and gait recognition for identifying the uniqueness of a person, together can establish a security link for mobile phone devices as an access control mechanism. Research has to the best of our knowledge not implemented these two technologies into one full system. What we will see in the next subsection is how we can apply activity and gait recognition approaches together and how this should work like.

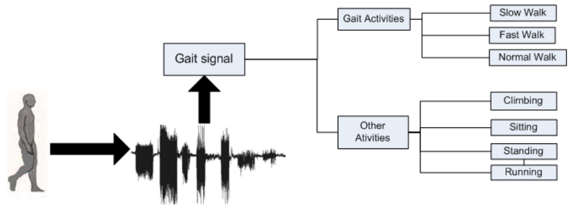

Fig. 4. Walking and Non-Walking Activities

B. Design and Proposal

The detection of everyday routine and leisure activities as we saw earlier like walking, running, sitting, and standing from gait signal recorded from wearable sensors make a step forward in the authentication. There has been done studies on gait recognition for authentication in mobile devices [76], but there are so far no studies in activity recognition for authentication. The data obtained from wearable sensors can be very useful for activity recognition as we have seen in the recent section. Therefore, activity recognition is becoming a necessary step regarding continues authentication that is based on gait using wearable motion recording sensors in mobile devices. A proposal towards full gait recognition includes activity recognition. This simply means that a full automatic system includes:

- **Activity Recognition** Identifying activities from a gait signal where we only focus on stable activities, such as walking normal, slow or fast.
- **Gait Recognition** Extraction of the unique from the stable walking activities to be used for authentication on a mobile device.

Since a full gait signal consist of different activities, we propose to divide the activity recognition in two phases. First phase is segmentation that is to find out where an each activities start and end point is located on the signal as illustrated in Figure 5. For this we propose the use of Sliding Windows, Top-Down, Bottom-Up and Sliding Window and Bottom-Up (SWAB) as referred to in section II-C1. Second phase is the classification where we can see which activities are useful to forward to the gait recognition mechanism as illustrated in Figure 6. The classification task as can be seen in Figure 6 consists in itself that pre-processing before inputting the data for segmentation, is needed. After the segmentation process we apply feature extraction approaches. Feature extraction is the process of extracting the most relevant information form the data segments. The features extracted then passes through the classification stage. This stage includes the classification process of the data and creation of classifiers which are used to identify different human activities. For the classifications there are different approaches to apply. We thus propose to apply methods that are shown in section II-C3

IV. Conclusion

In this paper we have proposed that by using activity recognition and gait recognition we can create a continuous and automatic authentication system on mobile devices.

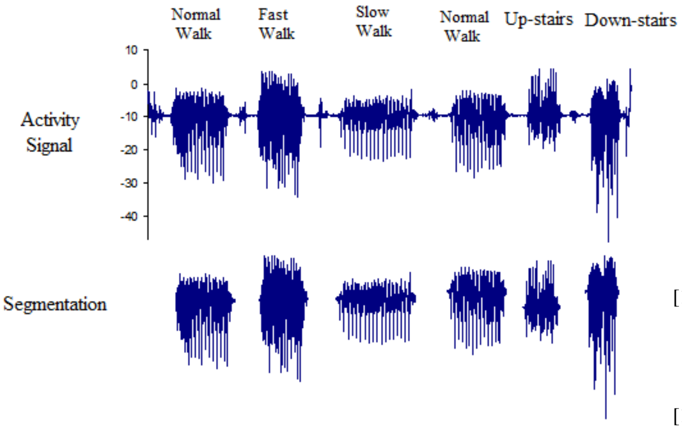

Fig. 5. A full gait signal without segmentation (upper signal figure) and segmented walks (lower signal figure)

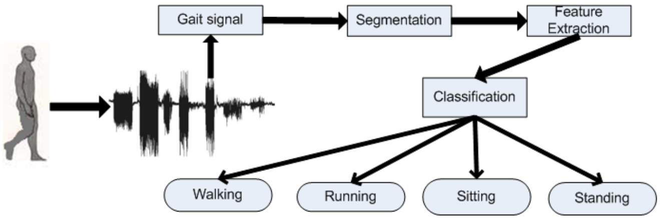

Fig. 6. Classification of the Activities

Since wearable sensor based gait recognition do not offer this mechanism today future work will then be to make an implementation of the design which was proposed during this paper. Activity and Gait Recognition has been studied separately in the recent years, but the interest has become so high lately when mobile phones today include these embedded accelerometers. The recognition accuracy for activity recognition has shown great results, which means to be useful for gait an automatic gait recognition system.

References

[1] Zhenyu He and Lianwen Jin. Activity recognition from acceleration data based on discrete consine transform and svm. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 5041 –5044, 2009.

[2] Duy Tâm Gilles Huynh. *Human Activity Recognition with Wearable Sensors*. PhD thesis, TU Darmstadt, Darmstadt, September 2008.

[3] Mohammad Omar Derawi. Accelerometer-based gait analysis, a survey. In *Norwegian Information Security Conference (NISK)*, NISK '10, pages 33–44, Gjøvik, Norway, 2010. Tapir.

[4] Ning Wang, Eliathamby Ambikairajah, Nigel H Lovell, and Branko G Celler. Accelerometry based classification of walking patterns using time-frequency analysis. *Conf Proc IEEE Eng Med Biol Soc*, 2007:4899–902, 2007.

[5] J R Kwapisz, G M Weiss, and S A Moore. Activity recognition using cell phone accelerometers. *Human Factors*, page 10–18, 2010.

[6] M.J. Mathie, N.H. Lovell, A.C.F. Coster, and B.G. Celler. Determining activity using a triaxial accelerometer. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 3, pages 2481 – 2482 vol.3, 2002.

[7] Xi Long, Bin Yin, and R.M. Aarts. Single-accelerometer-based daily physical activity classification. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 6107 –6110, 2009.

[8] L. Bao and S. S. Intille. Activity Recognition from User-Annotated Acceleration Data. *Pervasive Computing*, pages 1–17, 2004.

[9] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Predrag Klasnja, Karl Koscher, Anthony LaMarca, James A. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Rea, and Danny Wyatt. The Mobile Sensing Platform: An Embedded Activity Recognition System. *IEEE Pervasive Computing*, 7(2):32–41, April 2008.

[10] M. Ermes, J. Pärkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 12(1):20–26, January 2008.

[11] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A Practical Approach to Recognizing Physical Activities. In Kenneth Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing*, volume 3968 of *Lecture Notes in Computer Science*, chapter 1, pages 1–16. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006.

[12] Andreas Bulling, Jamie Ward, Hans Gellersen, and Gerhard Troester. Robust recognition of reading activity in transit using wearable electrooculography. In Jadwiga Indulska, Donald Patterson, Tom Rodden, and Max Ott, editors, *Pervasive Computing*, volume 5013 of *Lecture Notes in Computer Science*, pages 19–37. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-79576-6_2.

[13] Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, pages 113–116, Washington, DC, USA, 2006. IEEE Computer Society.

[14] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim. Human Activity Recognition via an Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis. pages 1–6, May 2010.

[15] Jie Yang, Shuangquan Wang, Ningjiang Chen, Xin Chen, and Pengfei Shi. Wearable accelerometer based extendable activity recognition system. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3641 –3647, May 2010.

[16] Mi hee Lee, Jungchae Kim, Kwangsoo Kim, Inho Lee, Sun Ha Jee, and Sun Kook Yoo. Physical activity recognition using a single triaxis accelerometer. In S. I. Ao, Craig Douglas, W. S. Grundfest, and Jon Burgstone, editors, *Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I, WCECS '09, October 20 - 22, 2009, San Francisco, USA*, Lecture Notes in Engineering and Computer Science, pages 14–17. International Association of Engineers, Newswood Limited, 2009.

[17] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 766–772, 2005.

[18] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, 3(4):50 – 57, 2004.

[19] Maja Stikic, Tâm Huynh, Kristof Van Laerhoven, and Bernt Schiele. ADL Recognition Based on the Combination of RFID and Accelerometer Sensing. In *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health 2008)*, page 258–263, Tampere, Finland, January 2008. IEEE Xplore, IEEE Xplore.

[20] Usman Naeem, John Bigham, and Jinfu Wang. Recognising activities of daily life using hierarchical plans. In *Proceedings of the 2nd European conference on Smart sensing and context*, EuroSSC'07, pages 175–189, Berlin, Heidelberg, 2007. Springer-Verlag.

[21] Jianfeng Chen, Alvin H. Kam, Jianmin Zhang, Ning Liu, and Louis Shue. Bathroom Activity Monitoring Based on Sound. pages 47–61. 2005.

[22] Emmanuel M. Tapia, Stephen S. Intille, and Kent Larson. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175–175. Springer Berlin / Heidelberg, 2004.

[23] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common Sense Based Joint Training of Human Activity Recognizers. In *In: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2237–2242, 2007.

[24] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-Sensor Activity Context Detection for Wearable Computing. In *In Proc. EUSAI, LNCS*, volume 2875, pages 220–232, 2003.

[25] J. Mäntyjärvi, J. Himberg, and T. Seppänen. Recognizing human motion with multiple acceleration sensors. volume 2, 2001.

[26] Pierre Barralon, Norbert Noury, and Nicolas Vuillerme. Classification of daily physical activities from a single kinematic sensor. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 3:2447–2450, 2005.

[27] Kristof Van Laerhoven and Hans-Werner Gellersen. Spine versus porcupine: A study in distributed wearable activity recognition. In *Proceedings of the Eighth International Symposium on Wearable Computers*, ISWC '04, pages 142–149, Washington, DC, USA, 2004. IEEE Computer Society.

[28] K. Van Laerhoven and O. Cakmakci. What shall we teach our pants? In *Wearable Computers, 2000. The Fourth International Symposium on*, pages 77 –83, 2000.

[29] Nishkam Ravi, Nikhil Dandekar, Prreetham Mysore, and Michael L. Littman. Activity Recognition from Accelerometer Data. *American Association for Artificial Intelligence*, 2005.

[30] C. Randell and H. Muller. Context awareness by analysing accelerometer data. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, volume 0, pages 175–176, Los Alamitos, CA, USA, 2000. IEEE Comput. Soc.

[31] Tanzeem Choudhury and Alex Pentland. Sensing and modeling human networks using the sociometer. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ISWC '03, pages 216–, Washington, DC, USA, 2003. IEEE Computer Society.

[32] NP Bidargaddi, L Klingbeil, and A Sarela. Detecting walking activity in cardiac rehabilitation by using accelerometer. *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 555–560, 2007.

[33] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 99–104, New York, NY, USA, 2005. ACM Press.

[34] M Sekine, T Tamura, M Akay, T Togawa, and Y Fukui. Analysis of acceleration signals using wavelet transform. *Methods Inf Med*, 39(2):183–5, 2000.

[35] Tâm Huynh, Ulf Blanke, and Bernt Schiele. Scalable recognition of daily activities with wearable sensors. In *3rd International Workshop on Location- and Context- Awareness (LoCA 2007)*, page 50–67, september 2007.

[36] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, 2007.

[37] Tâm Huynh and Bernt Schiele. Unsupervised discovery of structure in activity data using multiple eigenspaces. In *2nd International Workshop on Location- and Context- Awareness (LoCA 2006)*, Dublin, Ireland, May 2006.

[38] D.J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pages 44 – 51, 2005.

[39] Lucy Dunne, Pauline Walsh, Barry Smyth, and Brian Caulfield. Design and evaluation of a wearable optical sensor for monitoring seated spinal posture. *Wearable Computers, IEEE International Symposium*, 0:65–68, 2006.

[40] Kai Kunze, Michael Barry, Ernst A. Heinz, Paul Lukowicz, Dennis Majoe, and Jürg Gutknecht. Towards recognizing tai chi – an initial experiment using wearable sensors. In *Heinz1, Paul Lukowicz, Dennis Majoe, Jurg Gutknecht; Institute for Computer Systems and Networks UMIT*, 2006.

[41] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Tapia, and Stephen Intille. A Long-Term Evaluation of Sensing Modalities for Activity Recognition. In John Krumm, Gregory D. Abowd, Aruna Seneviratne, and Thomas Strang, editors, *UbiComp 2007: Ubiquitous Computing*, volume 4717 of *Lecture Notes in Computer Science*, chapter 28, pages 483–500. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[42] Andreas Krause, Technische U. München, and Asim Smailagic. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *In Proceedings of the 7th International Symposium on Wearable Computers*, pages 88–97, 2003.

[43] C. Zhu and W. Sheng. Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, PP(99):1 –5, 2011.

[44] Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature selection and activity recognition from wearable sensors. In *UCS*, pages 516–527, 2006.

[45] Sarah Brady, Lucy E. Dunne, Richard Tynan, Dermot Diamond, Barry Smyth, and Gregory M. P. O'Hare. Garment-based monitoring of respiration rate using a foam pressure sensor. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, pages 214–215, Washington, DC, USA, 2005. IEEE Computer Society.

[46] Nuria Oliver and Fernando Flores-Mangas. Healthgear: A real-time wearable system for monitoring and analyzing physiological signals. *Wearable and Implantable Body Sensor Networks, International Workshop on*, 0:61–64, 2006.

[47] Eamonn J. Keogh, Selina Chu, David Hart, and Michael J. Pazzani. An online algorithm for segmenting time series. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 289–296, Washington, DC, USA, 2001. IEEE Computer Society.

[48] Kristof Van Laerhoven and Ozan Cakmakci. What shall we teach our pants? In *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, ISWC '00, pages 77–, Washington, DC, USA, 2000. IEEE Computer Society.

[49] Shuangquan Wang, Jie Yang, Ningjiang Chen, Xin Chen, and Qinfeng Zhang. Human Activity Recognition with User-Free Accelerometers in the Sensor Networks. volume 2, pages 1212–1217, 2005.

[50] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recogn. Lett.*, 29:2213–2220, December 2008.

[51] Hassan Ghasemzadeh, Vitali Loseu, Eric Guenterberg, and Roozbeh Jafari. Sport training using body sensor networks: a statistical approach to measure wrist rotation for golf swing. In *Proceedings of the Fourth International Conference on Body Area Networks*, BodyNets '09, pages 2:1–2:8, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[52] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 159–163, New York, NY, USA, 2005. ACM.

[53] Clemens Lombriser, Nagendra B. Bharatula, Daniel Roggen, and Gerhard Tröster. On-body activity recognition in a dynamic sensor network. In *Proceedings of the ICST 2nd international conference on Body area networks*, BodyNets '07, pages 17:1–17:6, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[54] Ernst A. Heinz, Kai-steven Kunze, Stefan Sulistyo, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Experimental evaluation of variations in primary features used for accelerometric context recognition. In *in Proceedings of the 1st European Symposium on Ambient Intelligence (EUSAI 2003*, volume 1, pages 252–263, 2003.

[55] G. S. Chambers, S. Venkatesh, G. A. W. West, and H. H. Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *16th International Conference on Pattern Recognition*, volume 2, pages 1082–1085, 2002.

[56] Maja Stikic, Kristof Van Laerhoven, and Bernt Schiele. Exploring semi-supervised and active learning for activity recognition. In *Proceedings of the 12th International Symposium on Wearable Computers (ISWC 2008)*, pages 81–90, Pittsburgh, USA, September 2008. IEEE Press, IEEE Press.

[57] Oliver Amft, Mathias Stäger, and Gerhard Tröster. Analysis of chewing sounds for dietary monitoring. In *In UbiComp 2005*, pages 56–72. Springer, 2005.

[58] Seon W. Lee and Kenji Mase. Activity and Location Recognition Using Wearable Sensors. *IEEE Pervasive Computing*, 1(3):24–32, July 2002.

[59] Jamie A. Ward, Paul Lukowicz, Gerhard Troster, and Thad E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1553–1567, October 2006.

[60] Kristof Van Laerhoven, Nicky Kern, Hans-werner Gellersen, and Bernt Schiele. Towards a wearable inertial sensor network. In *In Proceedings of IEE Eurowearable*, pages 125–130, 2003.

[61] Felicity R. Allen, Eliathamby Ambikairajah, Nigel H. Lovell, and Branko G. Celler. An Adapted Gaussian Mixture Model Approach to Accelerometry-Based Movement Classification Using Time-Domain Features. pages 3600–3603, August 2006.

[62] E. A. Heinz, K. S. Kunze, M. Gruber, D. Bannach, and P. Lukowicz. Using Wearable Sensors for Real-Time Recognition Tasks in Games of Martial Arts - An Initial Experiment. *Computational Intelligence and Games, 2006 IEEE Symposium on*, pages 98–102, 2006.

[63] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer, 1st ed. 2007. corr. 2nd printing edition, January 2009.

[64] T. Huynh and B. Schiele. Towards less supervision in activity recognition from wearable sensors. In *Wearable Computers, 2006 10th IEEE International Symposium on*, pages 3 –10, 2006.

[65] C. M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

[66] Ceri Stefano Carey, Michael J. *Data-Centric Systems and Applications*. 2006.

[67] D. Minnen, T. Starner, M. Essa, and C. Isbell. Discovering Characteristic Actions from On-Body Sensor Data. *Wearable Computers, 2006 10th IEEE International Symposium on*, pages 11–18, 2006.

[68] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, pages 21–27. AAAI Press, 2005.

[69] Anthony Nguyen, Darren Moore, and Iain McCowan. Unsupervised clustering of free-living human activities using ambulatory accelerometry. *Conf Proc IEEE Eng Med Biol Soc*, 2007:4895–8, 2007.

[70] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference*, volume 6, pages 3037–3040, Washington, DC, USA, 1999. IEEE Computer Society.

[71] Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry Kautz. Opportunity Knocks: A System to Provide Cognitive Assistance with Transportation Services. pages 433–450. 2004.

[72] Dhruv Mahajan, Nipun Kwatra, Sumit Jain, Prem Kalra, and Subhashis Banerjee. A framework for Activity Recognition and Detection of Unusual Activities. In *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*, December 2004.

[73] Machine learning - who's the boss? http://awesomeful.net/posts/78-machine-learning-who-s-the-boss. [Online; accessed 13-march-2011].

[74] Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, pages 113–116, Washington, DC, USA, 2006. IEEE Computer Society.

[75] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler. Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):156–167, 2006.

[76] Mohammad Omar Derawi, Claudia Nickel, Patrick Bours, and Christoph Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. *Intelligent Information Hiding and Multimedia Signal Processing, International Conference on*, 0:306–311, 2010.