DEPARTMENT OF INFORMATICS ENGINNERING

SCHOOL OF ENGINEERING

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

MSC IN INFORMATICS & MULTIMEDIA

MASTER THESIS

HUMAN ACTIVITY RECOGNITION USING SMARTPHONES

THEODOROS E. MALLIOTAKIS

SUPERVISOR

Associate Professor Manolis Tsiknakis

HERAKLION

June 2015

Acknowledgments

I wish to express my sincere thanks to Dr. Manolis Tsiknakis, Associate Professor at the Department of Informatics Engineering of the Technological Educational Institute of Crete and supervisor of the present master thesis, for providing me with all the necessary facilities for the research. I am truly indebted and thankful to him for the inspiring guidance and continuous encouragement that profusely offered me.

Besides my supervisor, I would like to thank the members of the Biomedical Informatics Laboratory of the Department of Informatics Engineering at TEI Crete, George Vavoulas and Chariklia Chatzaki for providing me with all the instructions and all the necessary information about their previous work related to the MOBIFALL dataset.

My sincere thanks also go to Mathew Pediaditis, a member of the Computational Biomedicine Laboratory of the Institute of Computer Science at FORTH for his insightful comments and recommendations.

I also express my gratitude to my parents for the inexhaustible encouragement and support during this academic venture of mine.

Finally, I would like to thank Maria Datseri for her limitless patience and for her significant guidance on editing this master thesis.

June 2015,

Theodoros Malliotakis

Abstract

Many countries are facing an increasing elderly population. Consistently low birth rates and higher life expectancy is transforming the shape of the EU-28's age pyramid; probably the most important change will be the marked transition towards a much older population structure. As a result, the concept of "aging in place" has become an important problem to solve. To solve this problem, and allow the elderly to live a normal life within the community, the need arises for more intelligent infrastructures within the home. Specifically, there is a need for systems that can monitor the day to day tasks of the elderly and also recognize when something has gone wrong.

In monitoring a person's behavior, it can be useful to first establish normal patterns of behavior. In a home setting, this behavior can be decomposed into a number of essential activities, known as *Activities of Daily Living (ADL*s). The term "*activities of daily living*" refers to a set of common, everyday tasks, performance of which is required for personal self-care and independent living.

Human Activity Recognition (HAR) through computational methods is the process of identifying the actions and goals of one or more humans from a series of observations on the human's actions. Surveillance systems, human-computer interaction systems, ubiquitous smart health care systems and gaming systems are only a few examples of applications of HAR.

Existing approaches typically use vision sensors, inertial sensors or a mixture of both. Machine learning and threshold-based algorithms are often applied for the classification and recognition task.

The present thesis focuses on the study of HAR using smartphones. The main research question we are seeking to answer can be expressed as follows: "Which is the optimal setup to improve the performance of a HAR system implemented on a "smart" mobile device?" The main research question can be decomposed into a number or related sub-questions. The sub-questions can be expressed as a) *Which algorithms generate the best classification results, in terms of sensitivity and specificity?*, b) *Which is the optimal feature set to improve the performance of an implementation of a HAR system on mobile devices*? and c) *What are the factors and how they affect the performance of an implementation of a HAR system on mobile devices*?

In achieving answers to the above questions, we have selected a public data set, namely WISDM (Wireless Sensor Data Mining), which includes data collected from accelerometer sensors of mobile devices for six ADLs, namely: walking, jogging, (walking) up-the-stairs, (walking) down-the-stairs, sitting and standing. We focused on reproducing the most recent results published using this dataset. Then we have subsequently employed the MOBIFALL dataset, developed by the Biomedical Informatics Laboratory of the Department of Informatics Engineering at TEI Crete, which incorporates signals recorded from the accelerometer and gyroscope sensors for four different falls and nine different activities of daily living (ADLs).

The optimal set of features providing best performance was extensively explored, resulting in the minimal possible set of features that guarantee unambiguous identification of activities whilst on the other hand allowing for near real-time classification. A range of algorithms were also evaluated with respect to the sensitivity and specificity they provide. The results allow us to draw confident conclusions with respect to the best available configuration of algorithms, feature set and size of training set required for the implementation of a real time or near real time Human Activity Recognition system running on smartphones.

Keywords:

Human activity recognition, activities of daily living, smartphone, accelerometer.

Περίληψη

Πολλές χώρες ανά τον κόσμο αντιμετωπίζουν πρόβλημα με την αύξηση του πληθυσμού στις μεγάλες ηλικίες. Τα χαμηλά ποσοστά γεννητικότητας και η αύξηση του προσδόκιμου ζωής αλλάζουν το σχήμα της πυραμίδας του EU-28. Η πιο σημαντική αλλαγή θα είναι η μετατροπή της δομής του παγκόσμιου πληθυσμού σε μια γηραιότερη εκδοχή. Συνεπώς, η έννοια της γήρανσης του πληθυσμού έχει γίνει ένα σημαντικό πρόβλημα που χρήζει επίλυσης. Για να λυθεί το εν λόγω πρόβλημα και να επιτραπεί στους ηλικιωμένους να ζήσουν φυσιολογικά ως κομμάτι της κοινωνίας προκύπτει η ανάγκη για πιο ευφυής βοηθητικές υποδομές εντός του χώρου κατοικίας. Αυτές οι υποδομές – συστήματα πρέπει να μπορούν να παρακολουθούν τις καθημερινές εργασίες των ηλικιωμένων και με την κατάλληλη επεξεργασία των δεδομένων που τους δίνονται ως είσοδος να αναγνωρίζουν τα επικίνδυνα συμβάντα και ότι μπορεί να βλάψει τον παρακολουθούμενο.

Απαραίτητη για την αποτελεσματική παρακολούθηση του ατόμου και την ανάγνωση των μη φυσιολογικών συμπεριφορών είναι η δημιουργία αποδεκτών προτύπων συμπεριφοράς. Αυτά τα πρότυπα μπορούν να αναλυθούν περαιτέρω σε ακολουθίες από βασικές δραστηριότητες. Αυτές ονομάζονται «Δραστηριότητες Καθημερινής Ζωής (ΔΚΖ)». Αυτές οι δραστηριότητες είναι καθημερινές ενέργειες που είναι απαραίτητο να μπορούν να πραγματοποιηθούν από τον παρακολουθούμενο ώστε να μπορεί να παράσχει στον εαυτό του την καθημερινή φροντίδα που χρειάζεται και να ζήσει ανεξάρτητα.

Η «Αναγνώριση της Ανθρώπινης Συμπεριφοράς» ή «Αναγνώριση της Ανθρώπινης Δραστηριότητας» με χρήση υπολογιστικών μεθόδων είναι η διαδικασία προσδιορισμού των ενεργειών και των επιδιώξεων ενός ή περισσοτέρων ατόμων μέσα από μια ακολουθία παρατηρήσεων σχετικών με τις ενέργειες που εκτελεί το άτομο. Συστήματα παρακολούθησης, συστήματα αλληλεπίδρασης ανθρώπου – μηχανής, έξυπνα συστήματα υγείας και περίθαλψης και συστήματα ηλεκτρονικών παιχνιδιών είναι μόνο μερικά παραδείγματα όπου βρίσκει εφαρμογή η «Αναγνώριση Ανθρώπινης Δραστηριότητας (ΑΑΔ)».

Στη βιβλιογραφία αλλά και στα πραγματικά συστήματα που χρησιμοποιούνται από τον άνθρωπο, συναντώνται υλοποιήσεις της «Αναγνώρισης Ανθρώπινης Δραστηριότητας» που χρησιμοποιούν αισθητήρες όρασης, αισθητήρες αδράνειας καθώς και συνδυασμό

αυτών. Αλγόριθμοι μηχανικής μάθησης αλλά και αλγόριθμοι κατωφλίωσης χρησιμοποιούνται συχνά για «ταξιθέτηση» των ενεργειών και κατά συνέπεια τη αναγνώριση της ανθρώπινης δραστηριότητας.

Η παρούσα εργασία επικεντρώνεται στη μελέτη συστημάτων αναγνώρισης της ανθρώπινης δραστηριότητας με τη χρήση «έξυπνων» φορητών συσκευών. Το κύριο ερευνητικό ερώτημα το οποίο καλούμαστε να απαντήσουμε είναι: «Ποιο είναι η βέλτιστη πειραματική διάταξη η οποία βελτιώνει την απόδοση ενός συστήματος αναγνώρισης της ανθρώπινης δραστηριότητας;» Το ερευνητικό αυτό ερώτημα μπορεί να απαντηθεί αφού απαντηθούν επιμέρους ερωτήματα που προκύπτουν και είναι τα παρακάτω: α) Ποιοι αλγόριθμοι ταξιθέτησης ή ταξινόμησης παράγουν τα βέλτιστα αποτελέσματα σε ευαισθησία και ακρίβεια; β) Ποιο είναι το βέλτιστο σύνολο χαρακτηριστικών που εξάγονται από το αρχικό σήμα και βελτιώνει την απόδοση ενός συστήματος αναγνώρισης της ανθρώπινης δραστηριότητας; και γ) Ποιοι είναι γενικά οι παράγοντες και πώς αυτοί μπορούν να επηρεάσουν την υλοποίηση ενός συστήματος αναγνώρισης της ανθρώπινης δραστηριότητας σε μία «έξυπνη» φορητή συσκευή;

Με σκοπό να απαντηθούν τα ερευνητικά ερωτήματα που τίθενται παραπάνω ήταν απαραίτητο να επιλεχθούν κάποιες βάσεις δεδομένων ή αλλιώς σύνολα δεδομένων που έχουν συλλεχθεί με χρήση «έξυπνων» φορητών συσκευών. Η πρώτη βάση δεδομένων που επιλέχθηκε και είναι διαθέσιμη δημοσίως ήταν η βάση δεδομένων του WISDM (Wireless Sensor Data Mining) που δημιουργήθηκε από το πανεπιστήμιο του Fordham της Νέας Υόρκης. Αυτό το σύνολο δεδομένων έχει συλλεχθεί με τη χρήση των αισθητήρων επιτάχυνσης ή αλλιώς επιταχυνσιόμετρα που περιλαμβάνουν οι φορητές συσκευές. Τα δεδομένα αυτά αφορούν τις παρακάτω έξι καθημερινές δραστηριότητες: περπάτημα, τρέξιμο, ανάβαση σκάλας, κατάβαση σκάλας, κάθισμα και ορθοστασία. Για το παραπάνω επιλεγμένο σύνολο δεδομένων επικεντρωθήκαμε στην αναπαραγωγή της πιο πρόσφατα δημοσιευμένης υλοποίησης που αφορά τη συγκεκριμένη βάση δεδομένων καθώς και των αποτελεσμάτων που υπάρχουν σε αυτή τη δημοσίευση. Στη συνέχεια χρησιμοποιήσαμε το επίσης δημοσίως διαθέσιμο σύνολο δεδομένων «MOBIFALL». Το MOBIFALL δημιουργήθηκε από το εργαστήριο Βιοϊατρικής Πληροφορικής του Τμήματος Μηχανικών Πληροφορικής του ΤΕΙ Κρήτης το οποίο περιλαμβάνει σήματα που έχουν καταγραφεί από το επιταχυνσιόμετρο και το γυροσκόπιο φορητών συσκευών. Αυτή η βάση δεδομένων περιλαμβάνει σήματα για τέσσερις διαφορετικές πτώσεις και εννέα διαφορετικές δραστηριότητες της καθημερινής ζωής.

Τα βέλτιστα (εξαγόμενα από τα αρχικά δεδομένα) χαρακτηριστικά τα οποία οδηγούν στην καλύτερη δυνατή απόδοση των αλγορίθμων αναγνώρισης διερευνήθηκαν διεξοδικά. Αυτά προήλθαν από τη σύγκριση των αποτελεσμάτων που επιτεύχθηκαν από τα παραπάνω επιλεγμένα σύνολα δεδομένων (WISDM, MOBIFALL). Τα χαρακτηριστικά αυτά ελέγχθηκαν και αξιολογήθηκαν ώστε να ευρεθεί το μικρότερο δυνατό σύνολο χαρακτηριστικών που εγγυάται το σαφή προσδιορισμό των εξεταζόμενων δραστηριοτήτων. Παράλληλα έγινε προσπάθεια ώστε να μπορεί να πραγματοποιηθεί αναγνώριση δραστηριοτήτων «σχεδόν» σε πραγματικό χρόνο.

Η σύγκριση των αποτελεσμάτων των αλγορίθμων που εφαρμόστηκαν στα παραπάνω σύνολα δεδομένων οδήγησαν στο να προσδιοριστεί η επίδραση του μεγέθους του συνόλου δεδομένων που χρησιμοποιείται κατά τη διαδικασία «εκπαίδευσης» των αλγορίθμων, στην ποιότητα των αποτελεσμάτων. Επίσης πέντε διαφορετικοί αλγόριθμοι «ταξιθέτησης» αξιολογήθηκαν με βάση την ευαισθησία και την ακρίβεια των αποτελεσμάτων που παρέχουν.

Τα αποτελέσματα που εξήχθησαν από την έρευνα, μας επιτρέπουν να καταλήξουμε σε συμπαγή συμπεράσματα λαμβάνοντας υπόψη και τις επιτρεπτές παραμετροποιήσεις των χρησιμοποιούμενων αλγορίθμων «ταξιθέτησης». Με την παραπάνω μεθοδολογία καταλήγουμε στη δημιουργία ενός συστήματος αναγνώρισης ανθρώπινων δραστηριοτήτων καθημερινής ζωής το οποίο θα είναι εφικτό να υλοποιηθεί για σε «έξυπνες» φορητές συσκευές.

Λέξεις κλειδιά:

Αναγνώριση Ανθρώπινης δραστηριότητας, Δραστηριότητες Καθημερινής Ζωής, Έξυπνες φορητές συσκευές, Επιταχυνσιόμετρο.

Table of contents

List of figures

List of Tables

1 INTRODUCTION

1.1 Background to the study

Many countries are facing an increasing elderly population. Consistently low birth rates and higher life expectancy will transform the shape of the EU-28's age pyramid; probably the most important change will be the marked transition towards a much older population structure and this development is already becoming apparent in several EU Member States[1]. Likewise in Australia, a combination of longer life expectancy and decreased birth rate is expected to see the percentage of population aged 70 or older increase from 9% in 2007 to 13% by 2021 and 20% by 2051 [1]. As people age their function decreases, with, for example, 67.5% of Australians aged 75 and over affected by some kind of disability [2], putting increasing pressure on care and support services. As a result, the concept of "aging in place" has become an important problem to solve.

Aging in place allows the elderly to remain and interact with the community that they are familiar with and gives them a continued sense of independence, while also easing the burden that is seen on the limited capacity home care institutions. To solve this problem, and allow the elderly to live a normal life within the community, the need arises for more intelligent infrastructures within the home. Specifically, there is a need for systems that can monitor the day to day tasks of the elderly and also recognize when something has gone wrong.

In monitoring a person's behavior, it can be useful to first establish normal patterns of behavior. In a home setting, this behavior can be decomposed into a number of essential activities, known as Activities of Daily Living (ADLs).

In recent years, for the reasons stated above, Human activity recognition (HAR) has evoked notable scientific interest due to its frequent use in proactive computing. Proactive computing is the technology designed to anticipate an individual's needs and to take action to meet the needs on their behalf [3]. HAR is the process of identifying the actions and goals of one or more humans from a series of observations on the human's actions. Surveillance systems, human-computer interaction systems, ubiquitous smart health care systems and gaming systems are only a few examples of applications of

[1] http://ec.europa.eu/eurostat/statistics-explained/index.php/Population_structure_and_ageing

HAR. The most important application for human beings is in health care systems. Such systems apply HAR to give crucial information about humans and their physical activities. The demands for understanding human activities have grown in the health-care domain, especially in elder care support, rehabilitation assistance, diabetes, and cognitive disorders [4]. A huge amount of resources can be saved if sensors can help caretakers record and monitor the patients all the time and report automatically when any abnormal behavior is detected. Human activity recognition has been studied extensively and researchers have proposed different solutions to attack the problem. Existing approaches typically use vision sensor, inertial sensor and the mixture of both. Machine learning and threshold-based algorithms are often applied. One or multiple cameras have been used to capture and identify body posture [5]. Multiple accelerometers and gyroscopes attached to different body positions are the most common solutions [6]. Approaches that combine both vision and inertial sensors have also been purposed [7]. Another essential part of all these algorithms is data processing. The quality of the input features has a great impact on the performance. Some previous works are focused on generating the most useful features from the time series data set [8]. The common approach is to analyze the signal in both time and frequency domain.

As stated, many sensor-based applications have been proposed in the literature over the past decade. In [9], [10] the systems referred are examples of wearable sensor-based approaches of HAR. In [11] the HAR system presented uses wireless body sensors, a smartphone and a desktop workstation. Furthermore, [12], [13], [14], [15], [16] are proposing systems which use smartphone sensors to recognize human activities.

Today, a variety of smartphones exist with advanced features, like access to the internet, touch screens, built-in cameras and accelerometers for user interface control and other smart functions. These, recently introduced, features have given new capabilities to the smartphones and led them to become the most popular gadgets.

More specifically, smartphones used nowadays have a substantial computing power and an ability to send and receive high amounts of data. The number of smartphone users worldwide shows an increasing tendency. According to data from the International Data Corporation (IDC)[2], vendors shipped a total of 334.4 million smartphones worldwide in the first quarter of 2015, up 16.0% from the 288.3 million units in first quarter of 2014 but

[2] http://www.idc.com/prodserv/smartphone-os-market-share.jsp

down by 11.4% from the 377.6 million units shipped in fourth quarter of 2014. Android dominated the market with a 78.0% share in the first quarter of 2015. Following the latest technical reports, we focused on systems which use smartphone sensors to recognize human activities. Figure 1.1 illustrates the share in unit shipments for the different smartphone operating systems.

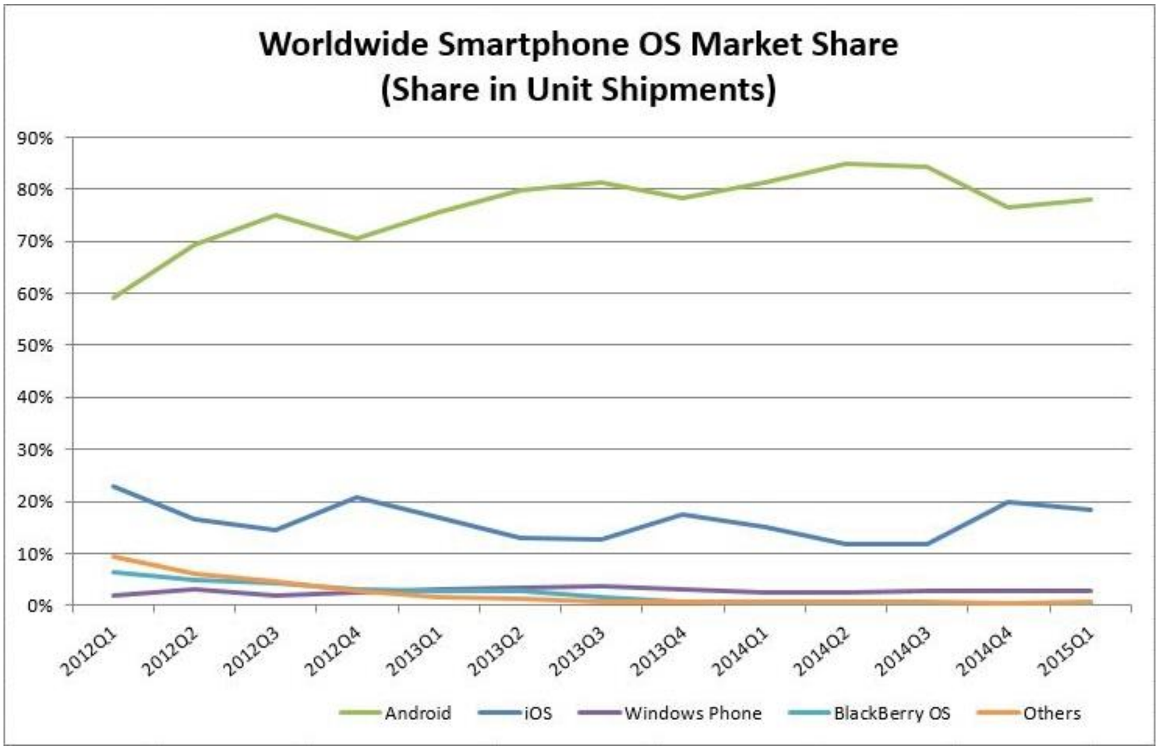


***Figure 1.1**: Worldwide smartphone OS market share in unit shipments (Picture taken from http://www.idc.com)*

1.2 The purpose of the study

The ultimate objective of the R&D work that led to the specific study and thesis is to develop a robust, smartphone based system for the accurate recognition of activities of daily life in real or near real-time.

As a result, the main research question we are seeking to answer in the context of the present master thesis can be expressed as follows: "Which is the optimal setup to optimize the performance of a HAR system implemented on a mobile device?"

The main research question can be decomposed into a number or related sub-questions. The sub-questions can be expressed as follows:

1) “Which algorithms generate the best classification results, in terms of sensitivity and specificity?”
2) “Which is the optimal feature set to maximize the performance of an implementation of a HAR system on mobile devices?”
3) “What are the factors and how they affect the performance of an implementation of a HAR system on mobile devices?”

1.3 Research methodology

In exploring answers to the above stated research questions, the following methodology has been adopted.

Initially we have selected the set of activities of daily life we would concentrate on (walking, jogging, upstairs, downstairs, sitting and standing). Subsequently a review of existing and available datasets was conducted, with the objective of selecting the one that is recognized, publically available and includes the selected activities. We then focused on reproducing the most recent results published using this dataset. In this process we have implemented a range of signal processing and feature extraction techniques. We have also used a range of classification algorithms, belonging to the machine learning category.

Once we have successfully reproduced the published results working with the WISDM dataset [12] we have repeated our analysis with the MOBIFALL dataset [16].

WISDM (Wireless Sensor Data Mining) includes data collected from accelerometer sensors of mobile devices for six ADLs, namely: walking, jogging, (walking) up-the-stairs, (walking) down-the-stairs, sitting and standing. MOBIFALL is a dataset, developed by the Biomedical Informatics Laboratory of the Department of Informatics Engineering at TEI Crete, which incorporates signals recorded from the accelerometer and gyroscope sensors for four different falls and nine different activities of daily living (ADLs). These datasets, described in detail in subsequent sections, both contain the above six ADLs which are the activities examined of this study.

In the process of our study we focused on estimating the optimal feature set used to correctly classify the activities. This was achieved by comparing the classification accuracies in the six examined activities while changing the feature set. Once we reached we believe is the optimal feature set for classification purposes, we analyzed a range of factors and their impact, namely: quality of dataset, the size of the training dataset, the window size and overlap. Finally, conclusions have been drawn and findings have been presented.

During this study all data (signals) were collected in android devices when humans were acting. In both datasets the android operating system was chosen for the collection of data, because it is free, open source, easy to program and is clearly the dominant OS in mobile devices marketplace. Regarding MOBIFALL, a Samsung Galaxy S3 smartphone with the LSM330DLC inertial module (3D accelerometer and gyroscope) was used to capture motion signals. Concerning WISDM, several types of android phones were employed including Nexus One, HTC Hero and Motorola Back-flip.

In addition, signals were processed in MATLAB (version 2013b) which is a software used for signal processing mainly for educational purposes. All the classifications were performed in WEKA 3.7.10 (Hall, Frank, Holmes, Pfahringer, Reutemann & Witten, 2009), a software which includes a large collection of classification algorithms.

1.4 Thesis Outline

In the first chapter we present a small description of this study. The second chapter includes a review of the literature focusing of analyzing similar systems which belong in the same scientific area.

Chapter 3 presents our experimental setup. In this chapter all evaluated datasets used for the comparison are described. We approach the activity recognition task as a supervised machine learning problem. The data collection process, the selection of datasets for comparison, the synchronization process of the selected datasets, the feature sets used in this investigation, the chosen set of activities and other details of the design of the experiment are presented in Chapter 3.

Chapter 4 elaborates on our implementation. All steps of this process are analyzed giving details useful for a possible reproduction of this study. The experiments and the results obtained are discussed in Chapter 5.

Chapter 6 is a presentation of the proposed system. All features included and best results of our investigation are discussed. Finally, chapter 7 gives our conclusions and discusses future directions.

2 LITERATURE REVIEW

ADLs can be assessed by different methods. An overview of these methods is given by Warren et al. [17]. Self-reports like questionnaires and activity diaries are a widely used tool to assess physical activity. They provide physical activity data from a large number of people in short time. However, self-reports induce problems with reliability, validity and sensitivity [18]. Therefore, the current trend is to replace self-reports with automatic ADL classification based either on vision based approaches, on small and light-weight wearable sensors like inertial measurement units or on sensors embedded on the modern smartphones.

In the subsequent sections we provide an elaborate review of current state-of-the-art of these different methods that have been employed for the purpose of automatically detecting and classifying Activities of Daily Life.

2.1 Vision-based approaches

Human activity recognition is an important area of computer vision research. Its applications include surveillance systems, patient monitoring systems, and a variety of systems that involve interactions between persons and electronic devices such as human-computer interfaces. Most of these applications require an automated recognition of high-level activities, composed of multiple simple (or atomic) actions of persons. An excellent recent review article provides a detailed overview of various state-of-the-art research papers on human activity recognition [19]. The authors propose an approach-based taxonomy that compares the advantages and limitations of each approach. In what follows we provide a short review of both the vision-based methodologies developed for simple human actions and those for high-level activities.

Vision-based human activity recognition is the process of classifying image sequences, included in a video into general activity categories. Usually, simple action verbs are used to label images. This problem has applications in domains such as visual surveillance, video retrieval, human-computer interaction, user interface design and machine learning. Due to the importance of its applications the field became of great interest and researchers proposed many methods which solve this classification problem.

The most common approach in vision-based human activity recognition is to extract features from video and to issue a label using an action verb, as mentioned above.

Classification algorithms are usually based on training data which are a part of the used dataset. This learning process is followed by the main classification procedure. The most widely used datasets should be mentioned because of their significance while comparing different approaches. These are: KTH human motion dataset [20], Weizmann human action dataset [21], INRIA XMAS multi-view dataset [22], UCF sports action dataset [23], Hollywood human action dataset [24], crowded videos dataset [25] and Nazli Ikizler web images dataset [26]. All these datasets are available to researchers during the feature extraction process which in video-based systems is called image representation.

Image representations can be divided into two categories: global and local representations. Global representation appears when a person is localized first in the image using background subtraction or tracking. Then, the region of interest is encoded as a whole, which results in the image descriptor. On the other hand, we speak of local representation when spatial-temporal interest points are detected first, and local patches are calculated around these points. Then the patches are combined into a final representation. Finally, there is a less commonly used representation category which is application specific and directly motivated by the domain of human action recognition.

The next step in vision-based human action recognition is clearly the classification process. This process is undertaken in three different ways:

- Without modeling variations in time (direct classification),
- By modeling such variations (temporal state-space classification) and
- Without modeling the action at all (action detection classification).

When temporal domain is not our first priority while classifying image representations, we should use direct classification. In this process, all frames of an observed sequence are summarized into a single representation or action recognition is performed for each frame individually. Commonly used methods in achieving the above classification are dimensionality reduction [27] (embedding the space of image representations onto a lower dimensional space), k-nearest neighbor classification [28] (the most common label among the k closest training sequences is chosen) and discriminative classifiers [29] (separating two or more classes, rather than modeling them).

When the classification process requires modeling variations, the approach is called temporal state-space classification. The models used consist of states connected by edges. These edges model probabilities between states, and between states and observations. Each state gives us the action performance at a certain moment in time. Each observation corresponds to the image representation at a given time. Temporal state-space models are categorized to generative and discriminative ones. Generative models [30] learn a joint distribution over both observations and action labels. They thus learn to model a certain action class, with all its variations. In contrast, discriminative models [31] learn probabilities of the action classes conditioned on the observations. They do not model a class but rather focus on differences between classes.

Finally, if the classification process does not involve modeling the action, then it is tagged as action detection classification [32]. Action detection approaches neither do explicitly model the image representation of a person in the image nor model action dynamics. Rather, they correlate an observed sequence to the labeled.

As a result, limitations are pointed out and promising directions are identified to address these limitations. Most of the reported work is restricted to fixed and known viewpoints, which severely limits its applicability. The use of multiple view-dependent action models solves this issue, at the cost of increased training complexity. Recently, researchers have begun to address the recognition of actions from viewpoints for which there is no corresponding training data [33]. Concerning classification, generative state-space models such as HMMs can model temporal variations, but have difficulties distinguishing between related actions (e.g. jogging and walking). In this respect, discriminative graphical approaches are more suitable. In future work, the flexibility of the classifier with respect to adding or removing action classes from the repository will play a more important role. Regarding datasets, given the increasing progress of action recognition algorithms, larger and more complex datasets are required which should enable innovative research attempts in realistic settings.

2.2 Wearable sensor-based approaches

During the past decade body worn wearable sensors became widely available and the use of them became widespread in several academic and industrial domains. Researchers were able to procure such components and build body sensor networks (BSNs) used to a variety of applications which are classified to categories like personal health care monitoring systems [34], physical fitness assessment [35], context awareness [36].

Instead of other human activity recognition approaches, wearable sensor-based systems transcend to the point of being with people throughout the day, enabling continuously collecting human activity information. This is exactly the reason why such a technology is commonly used in healthcare systems, e.g. elderly people's supporting platforms.

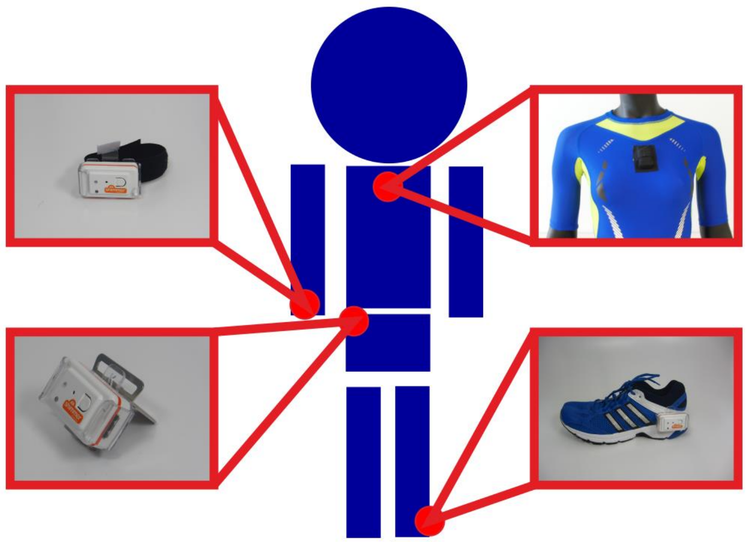


Figure 2.1: *Indicative wearable sensor placement. Sensor nodes are placed on the wrist, chest, hip, and ankle. (Picture taken from [9])*

Mannini and Sabatini [37] provide an overview of state-of-the-art human physical activity classification systems. Most of the approaches used accelerometers but differed in:

- number of sensor axes (uniaxial, biaxial and triaxial accelerometer),
- number of sensors and sensor placement,
- sampling rate,
- number of subjects,
- computed features,
- epoch/window size, and
- number and type of activities.

Regarding all these differences, it is difficult to compare newly proposed methods to existing approaches in the literature.

In considering human activity recognition using wearable sensors, we are faced with a task divided in two main stages: training and testing. The *training* stage is commonly based on a time series dataset of measured attributes which are collected while persons are performing each activity that HAR system claims to recognize. Such time series are split into time windows to apply feature extraction after filtering unnecessary information of the signals. Then, a learning method is used to create an activity recognition model from the dataset of features. Similarly, while *testing* a HAR system, data are collected in a time window and are used to extract features. These features are evaluated by a learning model in order to assign a label to the studied activity. Figure 2.2 shows the typical dataflow for HAR systems based on wearable sensors.

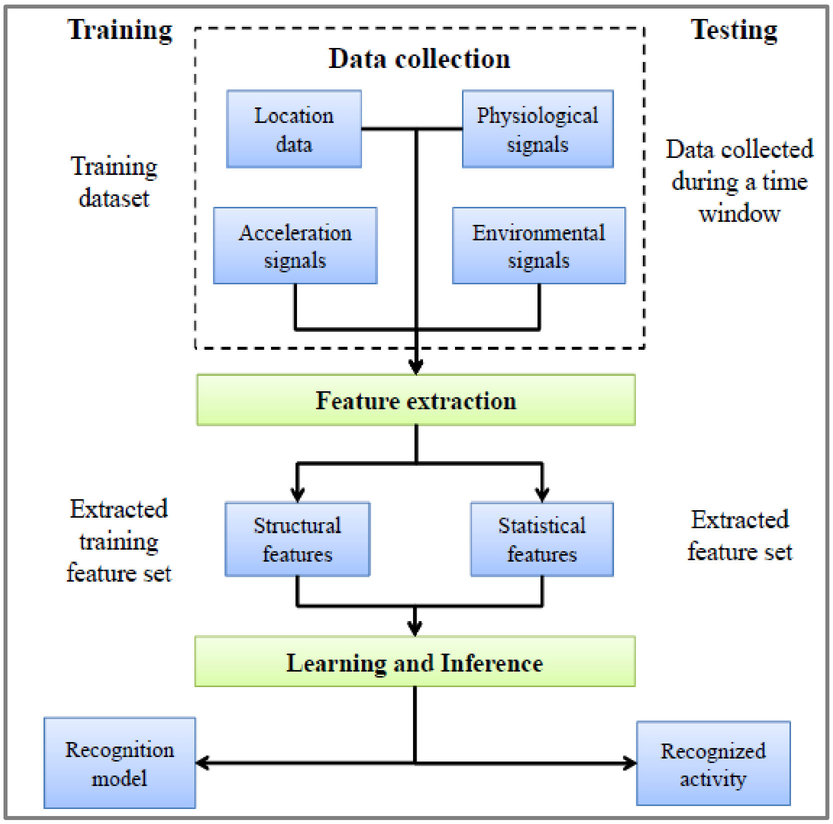


***Figure 2.2**: Dataflow for HAR systems based on wearable sensors, (Picture taken from [38])*

Figure 2.3 illustrates the generic data acquisition architecture for HAR systems based on wearable sensors. Firstly, sensors are attached to the person's body to measure attributes of interest, such as motion, location, temperature, ECG, sound, respiratory effort, oxygenated hemoglobin in the blood among others [9], [10], [37], [39], [40]. These sensors communicate with an integration device (ID), which can be a cellphone [11], [41], a PDA [40], a laptop [10] or a customized embedded system [42]. The main purpose of the ID is to preprocess the data received from the sensors and, in some cases, send them to an application server for real time monitoring, visualization, and/or analysis [43]. The communication protocol might be UDP/IP or TCP/IP, according to the desired level of reliability

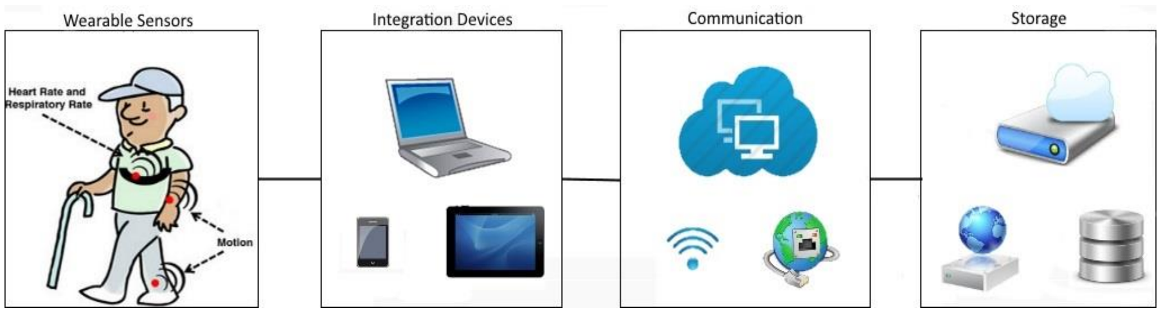


***Figure 2.3**: Data acquisition architecture of HAR systems based on wearable sensors*

2.3 Smartphone sensor-based approaches

Propelled by the race between Apple and Samsung to enhance their mobile products with cutting-edge sensor technology, the market for sensors in cellphones and tablets is set to nearly triple from 2012 through 2018, according to IHS Technology[3].

Worldwide market revenue for sensors used in mobile handsets and media tablets will rise to $6.5 billion in 2018, up from $2.3 billion in 2012. The fastest-expanding portion of the mobile sensor segment will be emerging devices, whose revenue will surge to $2.3 billion in 2018, up from just $24 million in 2012. In 2013, this segment posted dramatic growth, with revenue rising to more than $500 million[3].

Smartphones are bringing up new research opportunities for human-centered applications, where the user is a rich source of context information and the phone is the first-hand sensing tool. Latest devices come with embedded built-in sensors such as microphones, dual cameras, accelerometers, gyroscopes, etc. The use of smartphones with inertial sensors is an alternative solution for HAR. These mass-marketed devices provide a flexible, affordable and self-contained solution to automatically and unobtrusively monitor ADLs.

As a result, the developments previously mentioned coupled with advances in ubiquitous and pervasive computing have resulted in the development of a number of sensing technologies for capturing information related to physical activities of humans.

[3]https://technology.ihs.com/514244/apple-and-samsung-drive-adoption-of-next-generation-sensors

There have been a number of studies similar to the one proposed in the thesis that use commercially available mobile devices to collect data for activity recognition. Kwapisz et. al. [12] use an Android-based smart phone for recognizing very simple activities such as walk, jog, climb up and down the stairs, sit and stand. Yang [44] developed an activity recognition system using the Nokia N95 cell phone for distinguishing between different locomotion. Brezmes et. al. [45] proposes a subject dependent real time activity recognition system again using the Nokia N95 smart phone. Hache et. al. [46] use an accelerometer integrated with a blackberry Bold 9000 platform for detecting changes in the state of the subject caused by starting/stopping and postural changes in activities. Khan et. al. [14] use kernel discriminant analysis for recognizing very simple activities such as walking, up and down the stairs, running and resting on data collected from Samsung Omnia. Zhang et. al. [47] use an HTC smart phone for recognizing again simple activities using a support vector machines.

In the following sections we provide a short overview of the characteristics of the various sensors embodied in modern smartphones, which form the basis for all these exciting developments.

2.3.1 Smartphone sensors overview

As Figure 2.2 shows, modern smartphones carry a set of sensors very useful for many applications. Commonly, any modern smartphone is equipped with an accelerometer, a gyroscope, a magnetometer, a microphone and a digital camera. These sensors and their characteristics are mentioned in this section.

2.3.1.1 Accelerometer

An accelerometer measures proper acceleration, which is the acceleration it experiences relative to freefall and is the acceleration felt by people and objects. To put it in another way, at any point in space-time the equivalence principle guarantees the existence of a local inertial frame, and an accelerometer measures the acceleration relative to that frame. Such accelerations are popularly measured in terms of g-force.

An accelerometer at rest relative to the Earth's surface will indicate approximately 1 g (9.81 m/s^2) upwards, because any point on the Earth's surface is accelerating upwards relative to the local inertial frame (the frame of a freely falling object near the surface). To obtain the acceleration due to motion with respect to the Earth, this "gravity offset" must be subtracted and corrections should be made for effects caused by the Earth's rotation relative to the inertial frame. In our investigation this offset is not removed.

Single and multi-axis models of accelerometer are available to detect magnitude and direction of the proper acceleration (or g-force), as a vector quantity, and can be used to sense orientation (because direction of weight changes), coordinate acceleration (so long as it produces g-force or a change in g-force), vibration, shock, and falling in a resistive medium (a case where the proper acceleration changes, since it starts at zero, then increases). Micro-machined accelerometers are increasingly present in portable electronic devices and video game controllers, to detect the position of the device or provide for game input[4].

2.3.1.2 Gyroscope

A gyroscope is a device for measuring or maintaining orientation, based on the principle of preserving angular momentum. Mechanical gyroscopes typically comprise a spinning wheel or disc in which the axle is free to assume any orientation. Although the orientation of the spin axis changes in response to an external torque, the amount of and the direction of change are smaller and in a different direction than it would be if the disk were not spinning. When mounted in a gimbal (which minimizes external torque), the orientation of the spin axis remains nearly fixed, regardless of the mounting platform's motion[5]. In smartphones gyroscopes are implemented as MEMS (Micro Electrical Mechanical System). These are solid-microchip-packaged circuits which simulate the functionality of mechanical gyroscopes.

2.3.1.3 Magnetometer

Magnetometers are measurement instruments used for two general purposes: to measure the magnetization of a magnetic material like a ferromagnet, or to measure the strength and, in some cases, the direction of the magnetic field at a point in space.

In recent years magnetometers have been miniaturized to the extent that they can be incorporated in integrated circuits at very low cost and find increasing use as compasses in consumer devices such as mobile phones and tablet computers[6].

[4] https://en.wikipedia.org/wiki/Accelerometer
[5] https://en.wikipedia.org/?title=Gyroscope
[6] https://en.wikipedia.org/wiki/Magnetometer

2.3.1.4 Microphone

Microphone is an acoustic-to-electric transducer or sensor that converts sound in air into an electrical signal. Microphones are used in many applications such as telephones, hearing aids, public address systems for concert halls and public events, motion picture production, live and recorder audio engineering, two way radios, megaphones, radio and television broadcasting, in computers for recording voice, speech recognition, VoIP, for non-acoustic purposes such as ultrasonic checking or knock sensors. Innovative systems like smartphones are equipped with MEMS microphones. They include a pressure-sensitive diaphragm etched directly into silicon chip, which is usually accompanied with integrated preamplifier. Often MEMS microphones have built in analog-to-digital converter (ADC) circuits on the same CMOS chip making the chip a digital microphone and so more readily integrated with modern digital products[7].

2.3.1.5 Digital camera

A digital camera is placed in smartphones to encode digital images and videos and store them for later reproduction. Digital cameras are incorporated into many devices ranging from PDAs and mobile phones (called camera phones) to vehicles. As humans, we get the most information we can from vision information collected by digital cameras. As a result, photos and videos captured by smartphone digital cameras are very popular media. A photo by a digital camera is a matrix of light intensity of each pixel and a video is a sequence of "photos" frames which are displayed in rapid succession at a constant rate.

[7] https://en.wikipedia.org/wiki/Microphone

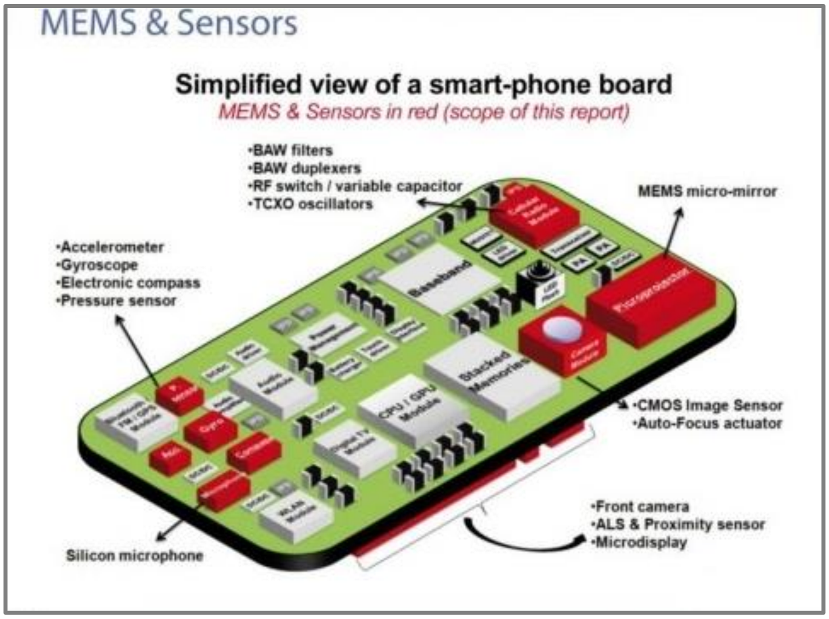


Figure 2.4: *Sensors of smartphones, (Picture taken from: [48])*

2.3.2 Smartphone sensor approaches

The urban landscape consists of entities like people, buildings, trees, vehicles and all the known environmental objects. As mentioned in [49] from Wazir Zada Khan et al. there is a general category which includes all the systems using smartphone sensors and is called urban sensing. If we wanted to define the urban sensing as an action, we would state that it is the deployment of sensors in order to collect data and then make decisions. Moreover, urban sensing is a general term commonly used but is not restricting the development of such systems out of the urban area. For example, sensing systems are developed in rural area or any other area the mobile phone communication exists. Urban sensing can be classified into two major classes in accordance to the awareness and involvement of people in the architecture of each system. The first is participatory sensing and the second is opportunistic sensing.

Participatory sensing: When people who are carrying everyday mobile devices act as nodes of the system, decide actively what applications requests to accept and provide data to the system e.g. give as input a photograph, the process is called participatory sensing. As a result, in such process use is directly involved.

Opportunistic sensing: On the other hand, when all the measurements are taken by the system without user interruption, the process is called opportunistic sensing. In such approaches user is not aware of active applications and is not involved at all.

In addition, when we discuss systems that are people-centric each of the above two classes includes three subclasses: personal, social and public sensing.

Personal sensing: Subject's information about his/her daily activities, health, personal and social contacts and location etc. are collected for further processing.

Social sensing: Subject's social information are collected and shared with subject's friends, social groups and communities.

Public sensing: Environmental data like noise, air pollution etc. and traffic information like free parking slots, traffic jam information and road bumps are collected and shared benefiting the general public.

In this section, we discuss the main characteristics of the above categories and mention some known mobile phone sensing systems which are included into those categories.

2.3.2.1 Personal participatory mobile phone sensing systems

1. *NeuroPhone:* Andrew T. Campbell et al [50] have proposed a system called NeuroPhone, using neural signals to control mobile phones for hands-free, silent and effortless human-mobile interaction.
2. *EyePhone:* EyePhone [51] is the first system capable of tracking a user's eye and mapping its current position on the display to a function/application on the phone using the phone's front-facing camera.
3. *SoundSense:* Hong Lu et al [52] have proposed a system called SoundSense, a scalable framework for modeling sound events on mobile phones.
4. *AndWellness:* John Hicks et al [53] have implemented a system called AndWellness which is a personal data collection system. It uses mobile phones to collect and analyze data from both active, triggered user experience samples and passive logging of onboard environmental sensors.
5. *SPA:* Kewei Sha et al [54] have proposed a smart phone assisted chronic illness self-management system (SPA), which greatly aided the prevention and treatment of chronic illness.

6. *BALANCE:* Tamara Denning et al [55] have proposed BALANCE (Bioengineering Approaches for Lifestyle Activity and Nutrition Continuous Engagement), a mobile phone-based system for long term wellness management.
7. *UbiFit Garden:* To address the growing rate of sedentary lifestyles, Sunny Consolvo et al [56] have developed a system, UbiFit Garden, which uses technologies like small inexpensive sensors, real-time statistical modeling, and a personal, mobile display to encourage regular physical activity.
8. *HyperFit:* P. Jarvinen et al [57] have proposed Hyperfit to develop communicational tools for personal nutrition and exercise management.
9. *PACER:* Chunming Gao et al [58] have proposed PACER which is a gesture-based interactive paper system that supports fine grained paper document content manipulation through the touch screen of a camera phone.
10. *Mobicare Cardio Monitoring System:* X Chen et al [59] have presented a cellular phone based online ECG processing system for ambulatory and continuous detection called Mobicare Cardio Monitoring System.

2.3.2.2 Social participatory mobile phone sensing systems

1. *CenceMe:* Emiliano Miluzzol et al [60] have presented a system called CenceMe, a personal sensing system that enables members of social networks to share their sensing presence with their buddies in a secure manner.
2. *DietSense:* Sasank Reddy et al [61] have developed a system DietSense to support the use of mobile devices for automatic multimedia documentation of dietary choices with just-in-time annotation, efficient post facto review of captured media by participants and researchers, and easy authoring and dissemination of the automatic data collection protocols.
3. *TripleBeat:* Rodrigo de Oliveira et al [62] have presented TripleBeat, a mobile phone based system that assists runners in achieving predefined exercise goals via musical feedback and two persuasive techniques: a glance able interface for increased personal awareness and a virtual competition.
4. *PEIR:* Min Mun et al [63] have presented the Personal Environmental Impact Report (PEIR) that uses location data sampled from everyday mobile phones to calculate personalized estimates of environmental impact and exposure.
5. *MoVi:* Xuan Bao and Romit Roy Choudhury [64] have built MoVi, a Mobile Phone based Video Highlights system using Nokia phones and iPod Nanos, and has

experimented in real-life social gatherings. MoVi is a collaborative information distillation tool capable of filtering events of social relevance.

2.3.2.3 Public participatory mobile phone sensing systems

1. EarPhone: *Rajib Kumar* Rana et al *[65]* have presented the design, implementation and performance evaluation of an end-to-end participatory urban noise mapping system called Ear-Phone.

2. *Micro-Blog:* Shravan Gaonkar et al [66] have presented the architecture and implementation system, called Micro-Blog. This system provides a reasonable location estimation avoiding the continuous use of GPS which is energy consumptive.
3. *V-Track:* Arvind Thiagarajan et al [67] have proposed a system called VTrack for travel time estimation using this sensor data and it addresses two key challenges: energy consumption and sensor unreliability. VTrack can use alternative, less energy-hungry but noisier sensors like WiFi to estimate both a user's trajectory and travel time along the route.
4. *TrafficSense:* Prashanth Mohan et al [68] have presented TrafficSense to monitor road and traffic conditions in a setting where there are much more complex varied road conditions (e.g., potholed roads), chaotic traffic (e.g., a lot of braking and honking), and a heterogeneous mix of vehicles (2-wheelers, 3-wheelers, cars, buses, etc.).
5. *LiveCompare:* Linda Deng et al [69] have presented LiveCompare that leverages the ubiquity of mobile camera phones to allow for grocery bargain hunting through participatory sensing.
6. *MobiShop:* Shitiz Sehgal et al [70] have proposed MobiShop, a novel people-centric application which facilitates sharing of product pricing information amongst consumers.
7. *Learmometer:* Laermometer [71] is developed to solve the problems of creating noise maps by utilizing mobile phones and their built-in microphones.
8. *MobGeoSen:* Eiman Kanjo et al [72] have developed a system called MobGeoSen which enables individuals to monitor their local environment (e.g. pollution and temperature) and their private spaces (e.g. activities and health) by using mobile phones in their day to day life.

9. *Citizen Journalist:* Citizen Journalist [73] is an application inspired by Micro-Blogs and involves participatory sensing, wherein PRISM provides location-based triggers to alert human users, who are in the vicinity of a location of interest, to respond to the application.
10. *Party Thermometer:* Party Thermometer [22] is an application which is also a human-query application, where queries are directed to users who are at parties. For example, a query could be how hot a particular party is. Like in the citizen journalist application, location is a key part of the predicate used to target the queries.

2.3.2.4 Personal opportunistic mobile phone sensing systems

1. *PerFallID:* Nicholas D Lane et al [74] have proposed PerFallID, utilizing mobile phones as a platform for pervasive fall detection system the only requirement of which is a mobile phone that has an accelerometer.
2. *I-Fall:* IFall [75] system is an alert system for fall detection using common commercially available electronic devices to both detect the fall and alert authorities.
3. *HealthGear:* HealthGear [76] is a real-time wearable system for monitoring, visualizing and analyzing physiological signals. HealthGear consists of a set of non-invasive physiological sensors wirelessly connected via Bluetooth to a cell phone which stores, transmits and analyzes the physiological data, and presents it to the user in an intelligible way.
4. *EmotionSense:* EmotionSense [77] is a mobile sensing platform for social psychological studies based on mobile phones. The key characteristics of this system include the ability of sensing individual emotions as well as activities, verbal and proximity interactions among members of social groups.
5. *Darwin:* Emiliano Miluzzo et al [78] have presented Darwin, an enabling technology for mobile phone sensing that combines collaborative sensing and classification techniques to reason about human behavior and context on mobile phones.

2.3.2.5 Social opportunistic mobile phone sensing systems

1. *WhozThat:* WhozThat [79] is a system that achieves the vision of seamless social interaction through MoSoNet technology by implementing a basic two-step protocol that first shares identities between any two nearby cellular smart phones (e.g., via Bluetooth or WiFi) and then consults an online social network with the

identity to import the relevant social context into the local context to enrich local human interaction.

2.3.2.6 Public opportunistic mobile phone sensing systems

1. *Nericell:* Nericell [68] is a system that performs rich sensing by piggybacking on smart phones that users carry with them in normal course. The authors have focused specifically on the sensing component, which uses the accelerometer, microphone, GSM radio, and/or GPS sensors in these phones to detect potholes, bumps, braking, and honking.

3 ACTIVITIES OF DAILY LIVING, DATASETS AND FEATURES USED FOR ACTIVITY RECOGNITION

In the following sections we initially discuss the issue of what activities are generally referred to as Activities of Daily Living and also introduce the concept of Instrumental Activities of Daily Living (IADLs). We subsequently review a number of publicly available datasets commonly used in published works for the recognition of daily activities. We conclude this chapter by selecting the daily activities of our focus and by presenting the features to be computed for the classification task.

3.1 Activities of daily living and instrumental activities of daily living

Activities of daily living (ADLs) are basic self-care tasks, akin to the kinds of skills that people usually learn in early childhood. ADLs are often mentioned by geriatric-care professionals in connection with Instrumental Activities of Daily Living (IADLs)[8], which are slightly more complex skills. ADLs are occasionally referred to as basic activities of daily living (BADLs).

Instrumental activities of daily living (IADLs) are the complex skills needed to successfully live independently. These skills are usually learned during the teenage years and include the following: Managing finances, handling transportation (driving or navigating public transit), shopping, preparing meals, using the telephone and other communication devices, managing medications, housework and basic home maintenance.

Together, ADLs and IADLs represent the skills that people usually need to be able to manage in order to live as independent adults, doctors, rehabilitation specialists, geriatric social workers, and others in senior care often assess ADLs and IADLs as part of an older person's functional assessment. Difficulty managing IADLs is particularly common in early Alzheimer's and other dementias. Assessing IADLs can help guide a diagnostic evaluation, as well as determine what kind of assistance an older person may need on a day-to-day basis.

[8] https://www.caring.com/articles/activities-of-daily-living-what-are-adls-and-iadls

In the context of current research efforts for developing robust solutions to the task of human activity recognition, several databases have been developed. Many of these databases, devoted to activity recognition and distress recognition, are distributed free of charge, for an academic and research use only, in order to be able to compare the results obtained.

In the following sections we present some of the most established such datasets, with the objective of selecting the most appropriate for our comparative analysis of results, which is the central task of the study reported in the current thesis.

3.2 DALIAC dataset

DALIAC [9] is an extensive and publicly available dataset of daily life activities (DLAs) which can be used as a benchmark for human activity recognition algorithms. DALIAC consists of data collected from wearable sensors placed on the participant's body. More specifically, sensor nodes were placed at four body positions: right hip, chest, right wrist and left ankle. Figure 1 shows sensors placement positions.

3.2.1 Equipment

Data were collected using four SHIMMER (Shimmer Research, Dublin, Ireland) sensor nodes. Shimmer sensor node contains a MSP430F1611 microcontroller. Each sensor node consisted of three accelerometer axes (A1, A2, A3) and three gyroscope axes (G1, G2, G3). The range of the accelerometer was ±6 g. The range of the gyroscopes was ±500 deg/s for the sensor nodes wrist, chest, hip and ±2000 deg/s for the sensor node on the ankle. The sampling rate was set to 204.8 Hz.

Apart from the sensor nodes, a mobile phone (Samsung Galaxy S2) was used as labeling device. An Android-based labeling App (running on the mobile phone) was used to label start time and end time of single activities concurrently to data collection.

The type of shirt and shoe (Figure 1) was the same for all participants. Four different shirt sizes (S, M, L, XL) were used in order to ensure tight fit and similar measurement conditions. To guarantee similar measurement conditions, the chest width of each volunteer were measured. Shirt sizes were assigned according to a size chart. The volunteers chose the shoe that they felt most comfortable in.

3.2.2 Activities

DALIAC contains inertial data collected from sensor nodes while 19 participants perform 13 daily life activities shown in table 3.1.

Table 3.1: *List of studied activities, abbreviations, durations, intensities and labels.*

ACTIVITY	DURATION (MIN)	INTENSITY (MET)	LABEL (L)
Sitting	1	1.3	1
Lying	1	1.0	2
Standing	1	1.3	3
Washing dishes	2	2.5	4
Vacuuming	1	3.3	5
Sweeping	1	3.3	6
Walking	n.a.*	3.5	7
Ascending stairs	n.a.**	5.0	8
Descending stairs	n.a.**	3.5	9
Treadmill running	2	9.0	10
Bicycling on ergometer(50 W)	2	3.5	11
Bicycling on ergometer(100 W)	2	6.8	12
Rope jumping	n.a.***	8.8	13

* All subjects had to walk on the university campus from one building to another.

** All subjects had to climb stairs to the third floor and then back again.

*** All subjects had to perform 5 trials with at least 5 jumps each.

3.2.3 Participants

While collecting data 19 healthy subjects participated in the study (8 of them were females and 11 males, aged 26 ±8 years, height 177 ±11 cm, weight 75.2 ±14.2 kg, mean ±standard deviation (SD).

3.2.4 File format

As a result of collecting data, 19 subsets were extracted and stored in files. Data is stored column by column separated by comma. The last column is the label of each activity as shown in table 3.1.

3.2.5 Feature Extraction

A generic feature set was defined for the classification systems BASE, HOUSE, WALK, and BICYCLE, which were computed for every sliding window. The generic feature set consisted of six features that were computed for every sensor axis and one feature that was computed for each of the accelerometer and gyroscope of each sensor node. The six features for every sensor axis included four time domain and two frequency domain features. The four time domain features were: *minimum amplitude, maximum amplitude, mean amplitude, and variance of amplitude*.

The minimum and maximum amplitude extracted range information of the amplitude. The mean and variance of the amplitude gave important knowledge about statistics of the signal. The two frequency domain features were: *spectral centroid and bandwidth.*

Spectral centroid and bandwidth delivered important information about the frequency distribution of the activities. The single feature that was computed for each sensor type (accelerometer or gyroscope) of one sensor node was the energy. The energy for each sensor type was calculated in three steps. First, the sum of the squared values for each axis was calculated. Second, the three sums were added together and divided by three. Third, this sum was divided by the number of samples. The energy gave important information about the activity level of a person. In total, this resulted in 152 features for each sliding window.

A different feature set was defined for the classification system REST. The gravitational component of the acceleration signal was extracted by a third-order elliptic low pass filter with an infinite impulse response and a cut-off frequency at 0.25 Hz. An important factor for the discrimination of the activities sitting, lying and standing was the orientation of the body.

3.3 UCI Dataset

UCI dataset [80] is a human activity recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. Collected data can be used as a benchmark for human activity recognition algorithms.

3.3.1 Equipment

The smartphone device that was used during the experimental process was Samsung Galaxy SII. It contains an accelerometer and a gyroscope for measuring 3-axial linear acceleration and angular velocity respectively at a constant rate of 50Hz, which is sufficient for capturing human body motion.

For AR purposes, a smartphone application was developed based on the Google Android Operating System. The experiments have been video-recorded to label the data manually.

3.3.2 Activities

Samsung Galaxy SII has accelerometer and gyroscope sensors embedded by default. This is a benefit that allowed researchers to classify a set of physical activities (standing, walking, laying, sitting, walking upstairs and walking downstairs) by processing inertial body signals through a supervised Machine Learning (ML) algorithm for hardware with limited resources.

3.3.3 Participants

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed the six activities mentioned previously wearing the smartphone on the waist. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data.

3.3.4 File format

For each record it is provided:

- Tri-axial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Tri-axial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.

The dataset includes the following files:

- 'README.txt'
- 'features_info.txt': Shows information about the variables used on the feature vector.
- 'features.txt': List of all features.
- 'activity_labels.txt': Links the class labels with their activity name.
- 'train/X_train.txt': Training set.
- 'train/y_train.txt': Training labels.
- 'test/X_test.txt': Test set.
- 'test/y_test.txt': Test labels.

The following files are available for the train and test data. Their descriptions are equivalent.

- 'train/subject_train.txt': Each row identifies the subject who performed the activity for each window sample. Its range is from 1 to 30.
- 'train/Inertial Signals/total_acc_x_train.txt': The acceleration signal from the smartphone accelerometer X axis in standard gravity units 'g'. Every row shows a 128 element vector. The same description applies for the 'total_acc_y_train.txt' and 'total_acc_z_train.txt' files for the Y and Z axis respectively.
- 'train/Inertial Signals/body_acc_x_train.txt': The body acceleration signal obtained by subtracting the gravity from the total acceleration.
- 'train/Inertial Signals/body_gyro_x_train.txt': The angular velocity vector measured by the gyroscope for each window sample. The units are radians/second.

3.3.5 Features

The features selected for this database come from the 3-axial accelerometer’s and gyroscope’s raw signals tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. Subsequently, they were filtered using a median filter and a 3rd order low-pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using an additional low-pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Also the magnitude of these 3-dimensional signals was calculated using the Euclidean norm: tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag.

Finally a Fast Fourier Transform (FFT) was applied to some of these signals producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag.

3.4 WISDM Dataset

The Wireless Sensor Data Mining (WISDM) project [12] involved the creation of a data collection that is connected with everyday human activities. The WISDM project’s goal was and still is to explore the research issues related to mining sensor data from powerful mobile devices and to build useful applications. To accomplish the experiment, android-based mobile devices and more specifically cellular phones were used. These devices are carrying an accelerometer sensor which is suitable to identify the activity the user is performing. Models of mobile devices are given in part 3.3.2. All experiments were performed at the Department of Computer and Information Science of Fordham University.

3.4.1 Equipment

The WISDM project's data set was generated using several models of android mobile phones. Three of them are Nexus one, HTC Hero and Motorola Back-Flip. All of them have an accelerometer and are also able to send data over the internet to the web server using a standard protocol.

Tri-axial accelerometers measure acceleration in all three spatial dimensions. These accelerometers are also capable of detecting the orientation of the device, due to the fact that they can also detect the direction of Earth's gravity, which can provide useful information for activity recognition.

The data acquisition was performed with the ACTITRACER activity recognition app, reported in the ACTITRACKER web page[9], which is available for free from the Google Play store, and which is continuously collecting new data.

3.4.2 Activities

In the WISDM study [12] there were examined six activities of daily living, namely: walking, jogging, ascending stairs, descending stairs, sitting, and standing. These activities were selected because they are performed regularly by many people in their daily routines. The activities also involve motions that often occur for substantial time periods, thus making them easier to recognize. Furthermore, most of these activities involve repetitive motions and this should also make the activities easier to recognize. While recording data for each of these activities, acceleration in three axes is stored in files. The z-axis captures the forward movement of the leg and the y-axis captures the upward and downward motion. The x-axis captures horizontal movement of the user's leg. Figure 3.2 demonstrates these axes relative to a user.

[9] https://actitracker.com/

3.4.3 Participants

In order to collect data for supervised learning task, it was necessary to have a large number of users carrying an Android-based smart phone while performing certain everyday activities. Before collecting these data, there was obtained an approval from the Fordham University IRB (Institutional Review Board) since the study involved "experimenting" on human subjects and there was a certain risk of harm (e.g., the subject could trip while jogging or climbing stairs). Then, twenty-nine volunteer subjects were asked to carry a smart phone while performing a specific set of activities.

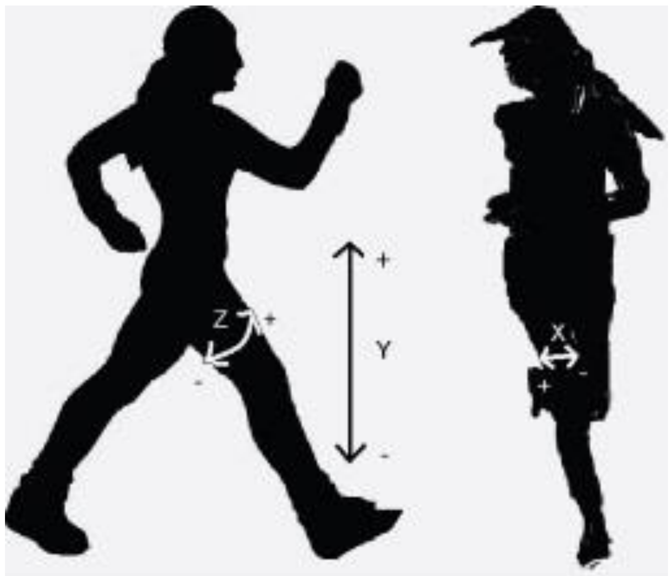


Figure 3.1: *Axes of motion relative to user (Picture taken from: [12])*

These subjects carried the Android phone in the front pocket of their pants and were asked to walk, jog, ascend stairs, descend stairs, sit, and stand for specific periods of time. The position or direction of the mobile device into the subjects' pockets was selected by each user in order to simulate better real world activities and record them.

3.4.4 File format

There are two different types of data collected during this effort: labeled and unlabeled data; the first is generated when the user optionally executes the app's training mode while the latter is captured otherwise. Since some users run this app throughout the day, the app has captured thousands of hours of unlabeled data.

Both of the two subsets are stored in files separated by comma and follow the format: [user], [activity], [timestamp], [x-acceleration], [y-acceleration], [z-acceleration]. This line is a representative example:

33, Jogging, 49105962326000, -0.6946377, 12.680544, 0.50395286.

3.4.5 Features

In all cases the accelerometer data were collected every 50ms, so the frequency was 20 Hz (20 samples per second). As it was mentioned, the example duration (ED) was 10s so there were 200 readings included in every 'window'. WISDM includes informative features based on these 200 raw accelerometer readings, where each reading contained an x, y, and z value corresponding to the three axes/dimensions (see Figure 2). A total of forty-three summary features (43) were generated, although these were all variants of just six basic features. The features are described below, with the number of features generated for each feature-type noted in brackets:

- Average (3): Average acceleration (for each axis)
- Standard Deviation (3): Standard deviation (for each axis)
- Average Absolute Difference (3): Average absolute difference between the value of each of the 200 readings within the ED and the mean value over those 200 values (for each axis)
- Average Resultant Acceleration (1): Average of the square roots of the sum of the values of each axis squared $\sqrt{(x_i^2 + y_i^2 + z_i^2)}$ over the ED
- Time Between Peaks (3): Time in milliseconds between peaks in the sinusoidal waves associated with most activities (for each axis)
- Binned Distribution (30): The range of values is determined for each axis (maximum – minimum), then divided into 10 equal sized bins, and then recorded what fraction of the 200 values fell within each of the bins.

3.5 The MOBIFALL dataset

MOBIFALL is a publicly available dataset which is targeted in collecting data from a smartphone when a participant is performing different types of falls[10]. This main purpose of creation and the large amount of fall data makes MOBIFALL a complete database suitable for fall detection.

[10] Available for download at http://www.bmi.teicrete.gr/index.php/research/mobifall

In addition, the MOBIFALL dataset includes data for activities of daily living (ADLs). These were selected following the next criteria. Firstly, activities which are fall-like are included. These are sequences where usually the subject finally stays motionless in different positions: sitting on a chair, step in a car, step out of a car. Secondly, activities which are sudden or rapid and are similar with falls like jumping and jogging. Finally, the MOBIFALL dataset contains the most common everyday activities, like walking, standing, ascending stairs and descending stairs. These activities are not registered with the objective of improving the primary objective of the dataset, i.e. fall detection; rather they were included as necessary for the gradual extension of the dataset towards recognition of complex everyday activities and, eventually, behaviors. Also, the fact that such activities are included is an advantage concerning the topic of '*Human Activity Recognition*' (HAR) in general. As a result, MOBIFALL dataset could be used when investigating HAR. All experiments related to the design of the protocol and acquisition of the MOBIFALL dataset were performed at the Technological Educational Institute of Crete.

3.5.1 Equipment

During the generation of the MOBIFALL dataset, data were recorded from the accelerometer and gyroscope sensors of a smartphone as well as orientation data; the orientation sensor is software-based and derives its data from the accelerometer and the geomagnetic field sensor. More specifically, a *Samsung Galaxy S3* smartphone with the *LSM330DLC inertial module (3D accelerometer and gyroscope)* was used to capture motion signals. The gyroscope was calibrated prior to the recordings using the device's integrated tool. Moreover, there was developed an Android application that records raw data for acceleration, angular velocity and orientation. A parameter that was enabled for providing the highest possible sampling rate was "SENSOR_DELAY_FASTEST". Each sample was stored along with its timestamp in nanoseconds.

3.5.2 Activities of Daily Living recorded

As already mentioned in *3.4.1* MOBIFALL includes fall activities and activities of daily living. Table 3.2 summarizes all the above and presents trials, duration and a short description for each activity.

3.5.3 Participants

For the purposes of MOBIFALL investigation 57 subjects (user ids 1-57) performed all the activities or a subset of them. In detail, 50 users completed successfully all ADL's and 55 users completed fall activities. More specifically, in activities of daily living just 10 files were removed during the editing process. These are: BSC_3_2, STU_10_2, STU_27_3, STU_56_1, SCH_21_4, SCH_51_4, STN_27_3, CSI_36_4, CSI_36_6, CSI_53_3. Filename follows the format: Activity_subjectid_trialno. All codes for activities are shown in table 3.4.3 1. Just for demographics, we have subjects aged between 20-47 years old. Also, the height of subjects is between 160 cm and 189 cm, the weight of them between 50 kilos and 120 kilos and finally 42 of them are males and 15 of them females.

Table 3.2: *Activities recorded in the MOBIFALL dataset, (Source: [16])*

Code	Activity	Trials	Dur	Description
FOL	Forward-lying	3	10s	Fall forward from standing, use hands to fall
FKL	Front-knees-lying	3	10s	Fall forward from standing, first on knees
SDL	Sideward-lying	3	10s	Fall sidewards from standing, bending legs
BSC	Back-sitting-chair	3	10s	Fall backward while trying to sit on a chair
STD	Standing	1	5m	Standing with subtle movements
WAL	Walking	1	5m	Normal walking
JOG	Jogging	3	30s	Jogging
JUM	Jumping	3	30s	Continuous jumping
STU	Stairs up	6	10s	Stairs up (10 stairs)
STN	Stairs down	6	10s	Stairs down (10 stairs)
SCH	Sit chair	6	6s	Sitting on a chair
CSI	Car-step in	6	6s	Step in a car
CSO	Car-step out	6	6s	Step out of a car

The majority of inertial sensor-based fall detection techniques require the sensor to be rigidly placed on the human body with a specific orientation. Usually a strap is used for this purpose. In contrast to this and in an attempt to simulate every-day usage of mobile phones, our device was located in trouser pockets freely chosen by the subject in any random orientation. For the falls, the subjects used the pocket on the opposite side of the direction of the fall. For the simulation of falls a relatively hard mattress of 5 cm in thickness (as used in martial arts) was employed to dampen the fall [16].

3.5.4 File format

The developed application uses a SQLite database in order to store activity types and each subject's personal information. The user of the application has the ability to insert, edit, or delete activities and participants. An automatic timer stops the data capture after the end of each trial. Three .txt-files are stored for each trial, one for the accelerometer data, one for the gyroscope data, and one for the orientation data. The header section of every file includes information about the recording, the subject, and the code of the activity performed. Then, there follows a tag (@DATA) and finally comes the stored data.

Each of the three '.txt' files has a data section. This section consists of four columns delimited with a comma (,) separator. The first column is same for all files and it is a timestamp in nanoseconds. It is the exact time in nanoseconds when the data were recorded. The rest of the columns differ depending on which file the reader reads.

- Accelerometer: timestamp (ns), x, y, z (m/s2)
- Gyroscope: timestamp(ns), x, y, z (rad/s)
- Orientation: timestamp(ns), azimuth, pitch, roll (degrees)

More specifically, the acceleration sensor gives device's acceleration for three axes x, y, z, the gyroscope sensor records the angular velocity and the orientation sensor gives device's angle in degrees around the three axes.

3.5.5 Features

Regarding the creation of MOBIFALL dataset, accelerometer and gyroscope data were collected. For the most of the features computed a value was extracted for each of the three axes x, y, z. In detail, the following features were computed:

- For each axis of the accelerometer and the gyroscope: mean, median, standard deviation, skew, kurtosis, minimum and maximum.
- For the three accelerometer axes: the slope SL within the time window defined as:

$$SL = \sqrt{} (max_x - min_x)^2 + (max_y - min_y)^2 + (max_z - min_z)^2$$

- For the tilt angle TAi between the gravitational vector and the y axis: mean, standard deviation, skew and kurtosis. The tilt angle is defined as:

$$TA_{i\,=}\, \sin^{-1}(y_i / \sqrt{}\, x_i^2 + y_i^2 + z_i^2)$$

where *x, y* and *z* is the acceleration in the respective axis.

- For the magnitude of the acceleration vector: mean, standard deviation, minimum, maximum, difference between maximum and minimum, entropy of the energy in 10 equal sized blocks (EEB), short time energy, spectral roll-off, spectral centroid and spectral flux.

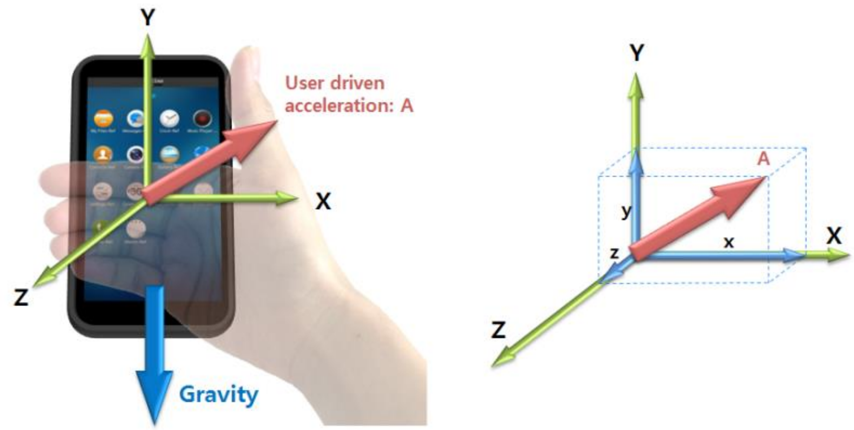


Figure 3.2*: Accelerometer data in 3 Dimensional space*

For all the features described above an absolute value is computed. Finally, no features were extracted from the orientation sensor, since this is a software-based sensor.

3.6 Selection of datasets for comparison

This research is focused on human activity recognition using data collected from smartphone sensors. Consequently, evaluation of datasets is needed in order to conclude with an optimal algorithm which recognizes human activities accurately.

As we can see from the general description of the DALIAC dataset, the DALIAC dataset cannot be a part of this comparison. The reason is that the DALIAC data are collected using accelerometer nodes statically placed on human body and without any use of smartphone. Furthermore, the compared datasets should include data in similar formats and with similar features which will be used for the classification process. UCI dataset differs from WISDM and MOBIFALL in file format. Also UCI data were recorded with a specific position for smartphone (waist mounted). As a result, UCI data cannot be compared to WISDM and MOBIFALL.

Therefore, we have chosen WISDM and MOBIFALL datasets for the comparison because they meet the circumstances. They include the same set of activities of daily living, namely *walking, jogging, upstairs, downstairs, sitting, and standing* and they use similar file formats. The position of the mobile device is not specific and it is up to the user as to what orientation it will be put into the pocket, which brings the experiment closer to the reality.

3.7 Synchronization of selected datasets

Synchronization is a requirement for the comparison of the selected datasets (MOBIFALL, WISDM). In this study there is chosen the WISDM dataset's file format and sampling rate as our investigation's constants because of the simplicity of the file format and the low value of sampling rate. As a result, it has been used WISDM file format and 20Hz sampling rate.

Considering the difference between MOBIFALL sampling rate (87Hz) and WISDM sampling rate (20HZ), MOBIFALL data was processed with the technique of linear interpolation. More specifically we report MATLAB code used for interpolation:

```
interpolaccx=interp1(acc.rel_time, acc.acc_x, rel_time_arr, 'linear');
interpolaccy=interp1(acc.rel_time, acc.acc_y, rel_time_arr, 'linear');
interpolaccz=interp1(acc.rel_time, acc.acc_z, rel_time_arr, 'linear');
```

The final signal of MOBIFALL which was used to extract features had a sampling rate of 20Hz. Moreover, all gyro features were removed because WISDM dataset does not include such values. Eventually, the unified file format used for the purposes of this experiment was as follows:

user_id, label, timestamp, acc_x, acc_y, acc_z

Example line: 1, downstairs, 2693088902000, 0.110, 9.190, 2.760

3.8 Feature sets

The target of this investigation is to create a HAR system which recognizes human activities accurately. This HAR system will include features from two existing feature sets (MOBIFALL, WISDM). For the simplicity of our description, the used feature sets are names as follows:

- The features reported in section 3.4.5 as MOBIFALL feature set (MFS) (no gyroscope features are included).
- The features mentioned in section 3.3.5 as WISDM feature set (WFS).

- The union of features of the two above sets as hybrid feature set (HFS).
- The optimal features occurring from all tests made as optimal feature set (OFS).

3.9 Investigated activities

As it is referred in section 3.5 the activities that participate in this study are: *walking, jogging, upstairs, downstairs, sitting and standing*. These are the most common everyday human activities. Apart from sitting and standing the rest of them might be characterized as periodic. These activities tend to include local maximums and minimums (peaks). Figures 3.3 to 3.8 show the plotted signals of all activities included in the MOBIFALL dataset. In addition, figures 3.9 to 3.14 present the plotted signals of all activities included in the WISDM dataset. All plots were performed in MATLAB (version 2013b).

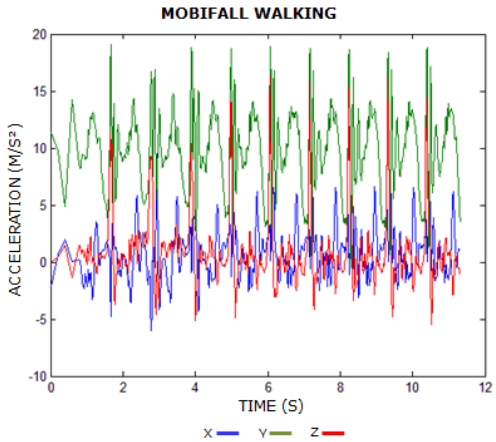


Figure 3.3: *Mobifall Walking*

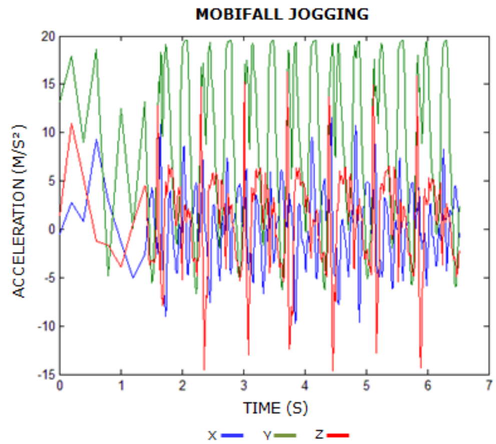


Figure 3.4: *Mobifall Jogging*

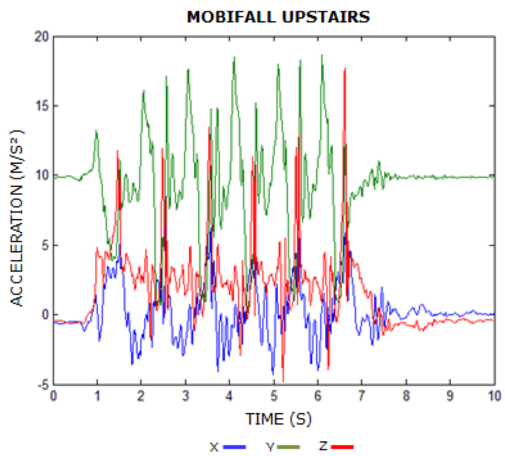


Figure 3.5: *Mobifall Upstairs*

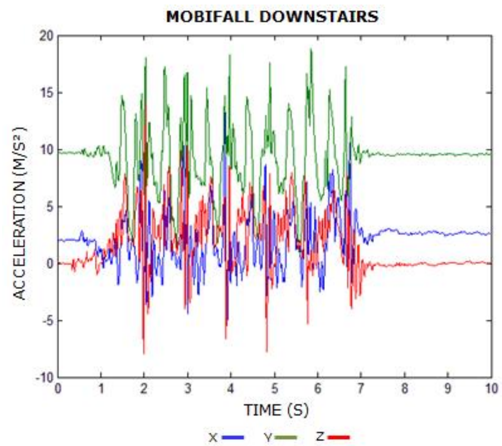


Figure 3.6: *Mobifall Downstairs*

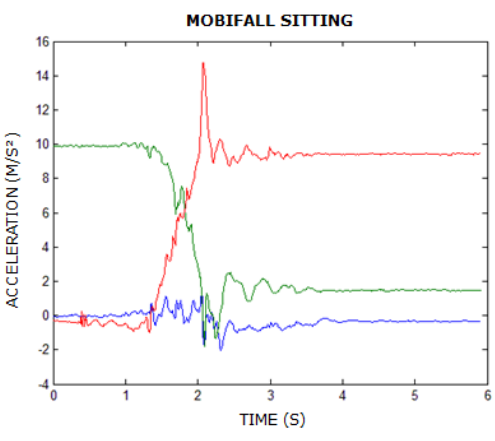


Figure 3.7: *Mobifall Sitting*

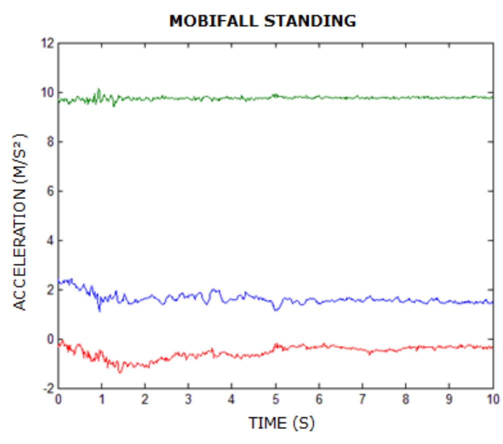


Figure 3.8: *Mobifall Standing*

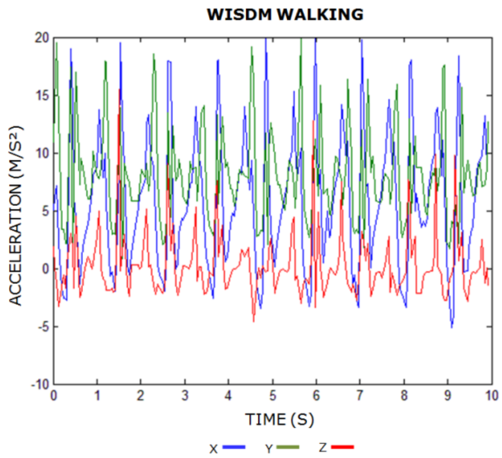


Figure 3.9: *WISDM Walking*

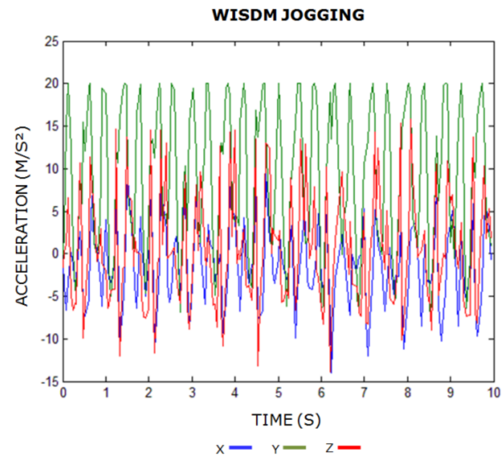


Figure 3.10: *WISDM Jogging*

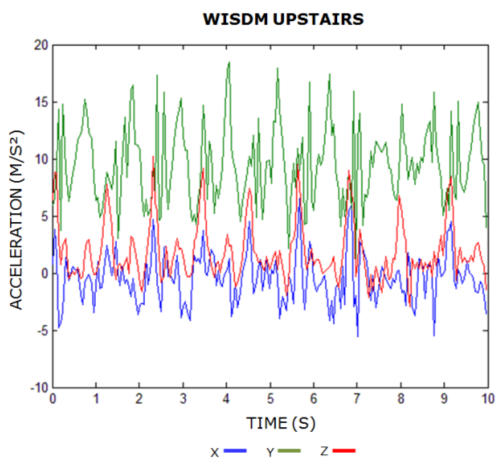


Figure 3.11: *WISDM Upstairs*

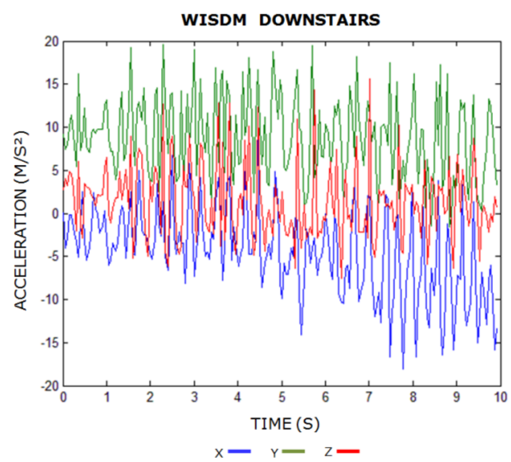


Figure 3.12: *WISDM Downstairs*

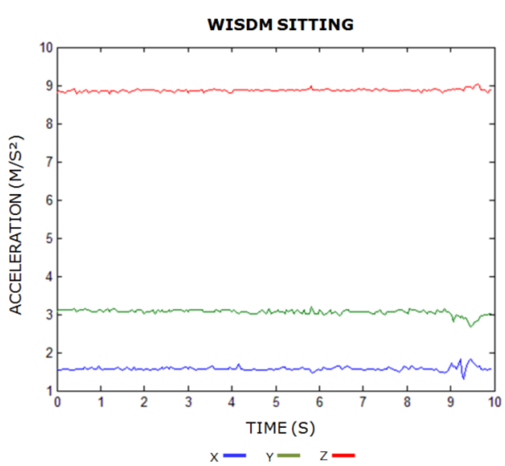


Figure 3.13: *WISDM Sitting*

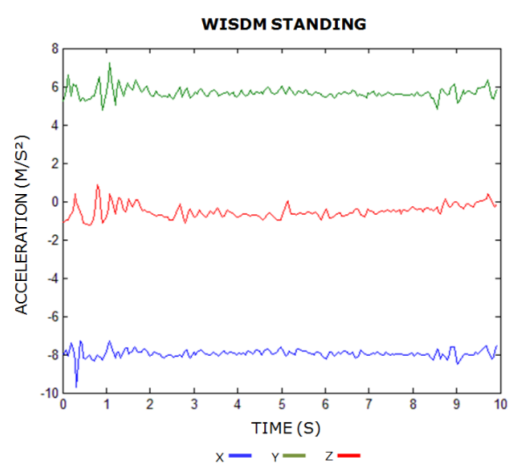


Figure 3.14: *WISDM Standing*

4 IMPLEMENTATION

4.1 Software used

During this study two basic software platforms were used. Concerning the programming part of this research, MATLAB for building functions and scripts needed to produce the necessary feature files. These feature files ('.csv') were the inputs to the second software platform used in this study which was WEKA (version 3.7.10). Moreover, MATLAB was used for all plots of signals e.g. figure 3.3.

4.1.1 MATLAB

MATLAB (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and a fourth-generation programming language. Developed by MATHWORKS, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages, including C, C++, Java and Python. It is a high-level language and includes interactive environment for numerical computation, visualization, and programming[11].

MATLAB can be used for a large number of applications, including signal processing and communications, image and video processing, control systems, and measuring of the computational cost and computational biology. More than one million engineers and scientists in industry and education use MATLAB.

The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java. More than a million engineers and scientists in industry and academia use MATLAB.

4.1.2 WEKA

WEKA (**W**aikato **E**nvironment for **K**nowledge **A**nalysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. WEKA is a software freely available, under the GNU General Public License. It is a collection of machine learning algorithms for data mining tasks[12]. The algorithms can either be applied directly to a dataset or called from your own Java code. WEKA contains tools for data pre-processing,

[11] http://www.mathworks.com/
[12] https://en.wikipedia.org/wiki/Weka_(machine_learning)

classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

WEKA supports several standard data mining tasks. More specifically, data preprocessing, clustering, classification, regression, visualization and feature selection. All of WEKA's techniques are predicated on the assumption that the data are available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). WEKA provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining, but there is separate software for converting a collection of linked database tables into a single table that is suitable for processing using WEKA. Another important area that is currently not covered by the algorithms included in the WEKA distribution is sequence modeling[13].

4.2 Different data formats and process sequences

As mentioned in part 3.5 we have chosen **WISDM** and **MOBIFALL** datasets for the purposes of this investigation. These are public datasets available in two different formats. WISDM dataset is given as a '.txt' file named 'WISDM_ar_v1.1_raw.txt' which includes all data for all users recorded. On the other hand, MOBIFALL dataset consists of a number of folders each of them named JOG (jogging) or WAL (walking) etc. depending on the activity which includes. Moreover, each folder contains many '.mat' files. Each '.mat' filename follows the format:

'activity_acc_subjectid_trialno_postfix.mat'

Concerning, WISDM dataset we should mention that there is a new version of data included in a file named 'WISDM_at_v2.0_raw.txt' containing more users than the first and more data generally. Furthermore, regarding MOBIFALL dataset there are more activities recorded like bicycling, stepping into car or stepping out of car. These facts might be at the center of a future investigation.

Accordingly to all the above, in our experiment there are two similar but not equivalent data process sequences. The first is MOBIFALL data process and the second is WISDM data process and both are described in sections 4.2.1 and 4.2.2.

[13] http://www.cs.waikato.ac.nz/ml/weka/

4.2.1 The sequence of processing for the MOBIFALL data

Step 1: As it is mentioned in part 4.2, the MOBIFALL data consists of a large number of '.mat' files classified into folders. The first step of processing such files (raw data) is to convert them into a unified file format for the purposes of this investigation. For the simplicity of this process, the file format referred in part 3.6 was chosen. Also, an interpolation of the raw values was necessary in order to convert the sampling rate to 20Hz. So, after the end of first step all '.mat' files were converted to data which are suitable for extracting the MOBIFALL features.

Step 2: The second step of data processing is extracting features. If MOBIFALL is the current feature set which is going to be produced, a MATLAB function is called which gets as input a simple '.mat' file and produces a '.csv' with the computed features as described in part 3.4.6. All gyroscope features are removed. During this step a crucial fact must be mentioned. Assuming that N denotes the number of samples included in an input file ('.mat'). If N is less than the current window size, the file is ignored as a signal noise file and the process continues. This significant point will be discussed in more detail in the Results Section.

Step 3: After the steps 1 and 2 a large number of '.mat' files is produced. Also the '.csv' files which include the computed features are generated. Note that the number of '.csv' files might not be equal to the number of '.mat' files because some files might be ignored as stated in step 2. Usually the number of files ignored is not large and there is no negative effect for the whole process. Nevertheless, all '.csv' files are concatenated in order to create the final '.csv' file which will be used for the classification. There is no specific order in which '.csv' files must be concatenated. The MATLAB function which is responsible for the concatenation opens the directory which contains '.csv' files and concatenates them with the default operating system order.

Step 4: Finally, a '.csv' file which includes all features is produced. This output file is the input for WEKA software which is used for the classification process.

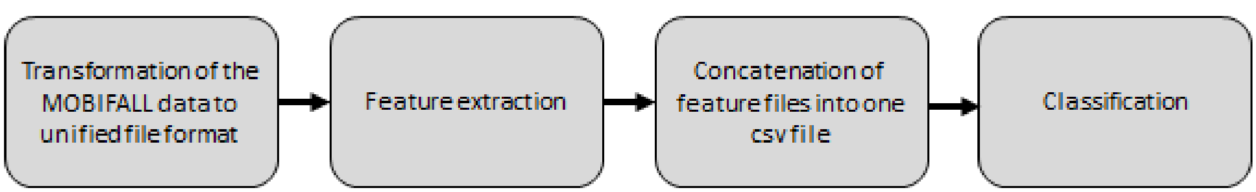


Figure 4.1*: The data handling, processing and analysis pipeline in MOBIFALL*

During this step we remove the columns user_id and trial_no since they are not significant for the classification. No WEKA filter is used and all classifications are done with algorithms IBK, J48, Logistic, Multilayer perceptron and LMT from WEKA's algorithm set. IBK algorithm is used for its simplicity and because it is the algorithm used in MOBIFALL implementation.

Moreover, J48, Logistic and Multilayer perceptron are tested because they are the algorithms used in WISDM implementation. Finally, LMT is the author's additional choice, due to fact that during the testing procedure the obtained results were very satisfactory. For all classifications, a 10 fold cross validation was used which is the default setting of WEKA software. Figure 4.1 represents the sequence of processing for the MOBIFALL data as it was described above.

4.2.2 The sequence of processing for the WISDM data

Step 1: As pointed out in part 4.2 WISDM data are included into a large '.txt' file named 'WISDM_ar_v1.1_raw.txt'. Therefore, it is necessary to split WISDM raw data into smaller files in order to divide data by user and activity. The produced files follow the format adverted in step 1 of part 4.2.1. Considering WISDM raw data, there is no need of interpolation because they are given already in 20Hz sampling rate. After this step, a large number of '.mat' files are created in order to be used in the next steps of the data processing.

Steps 2, 3, 4 are similar to MOBIFALL's data process sequence steps mentioned in part 4.2.1. It is noteworthy that in step 2, if an input file includes fewer samples than the window size it is ignored and the process continues. Figure 4.2 shows WISDM data process sequence.

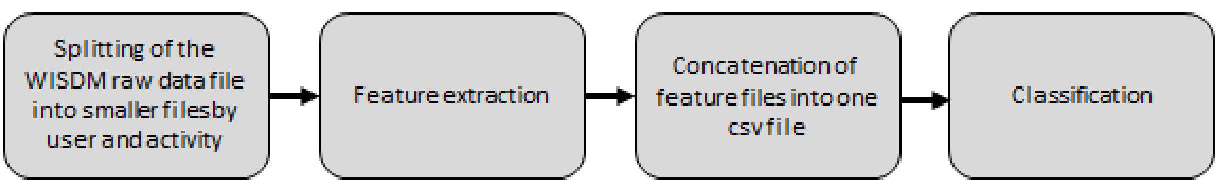


***Figure 4.2**: The data handling, processing and analysis pipeline in WISDM*

4.3 GUI implementation

During our exploratory activities and tests, because of the large scale of data which were processed, there was a need of an automated environment which could generate the results. This need led us to the creation of a simple GUI environment in MATLAB GUIDE. Figure 4.3 represents the implemented GUI.

Our GUI includes a radio button group in order to select which mode we want to run. There is also functionality present allowing for changing the frequency of the experiment by giving it as a parameter. Also, there is an edit box for the user to choose the delimiter separating the values into the raw '.txt' file (only if the file is a '.txt' like WISDM format), and edit boxes for setting the window size and the step to be used for the current experiment. Finally, a file or folder selection functionality enables the user to choose a file (e.g. 'WISDM_ar_v1.1_raw.txt') or a folder for example named 'data'. When the 'Start Process' button is pressed the process begins.

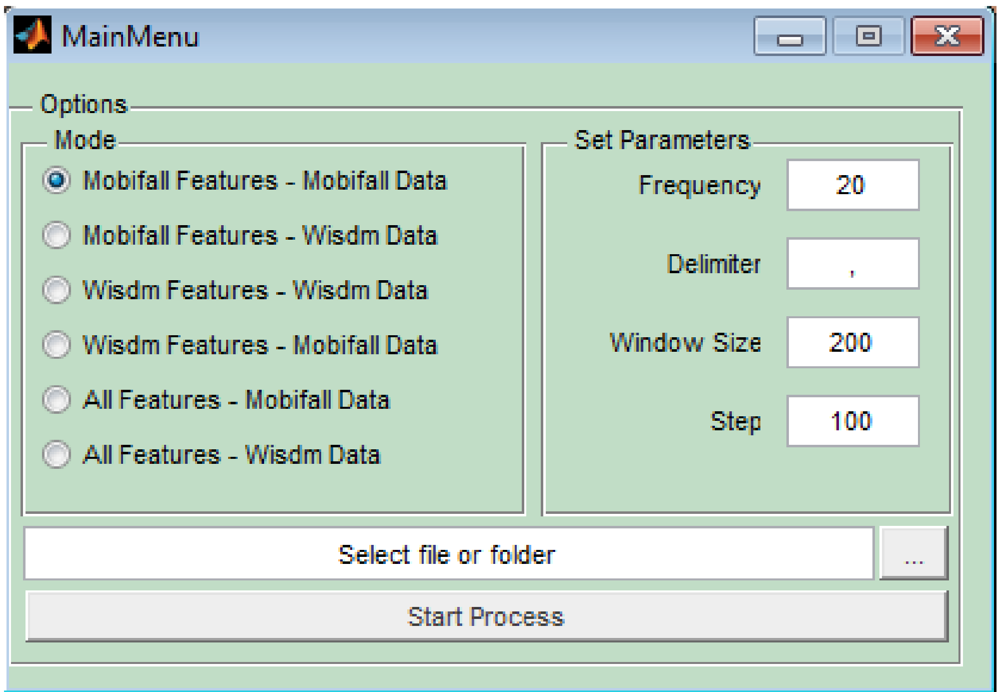


Figure 4.3: *GUI implementation*

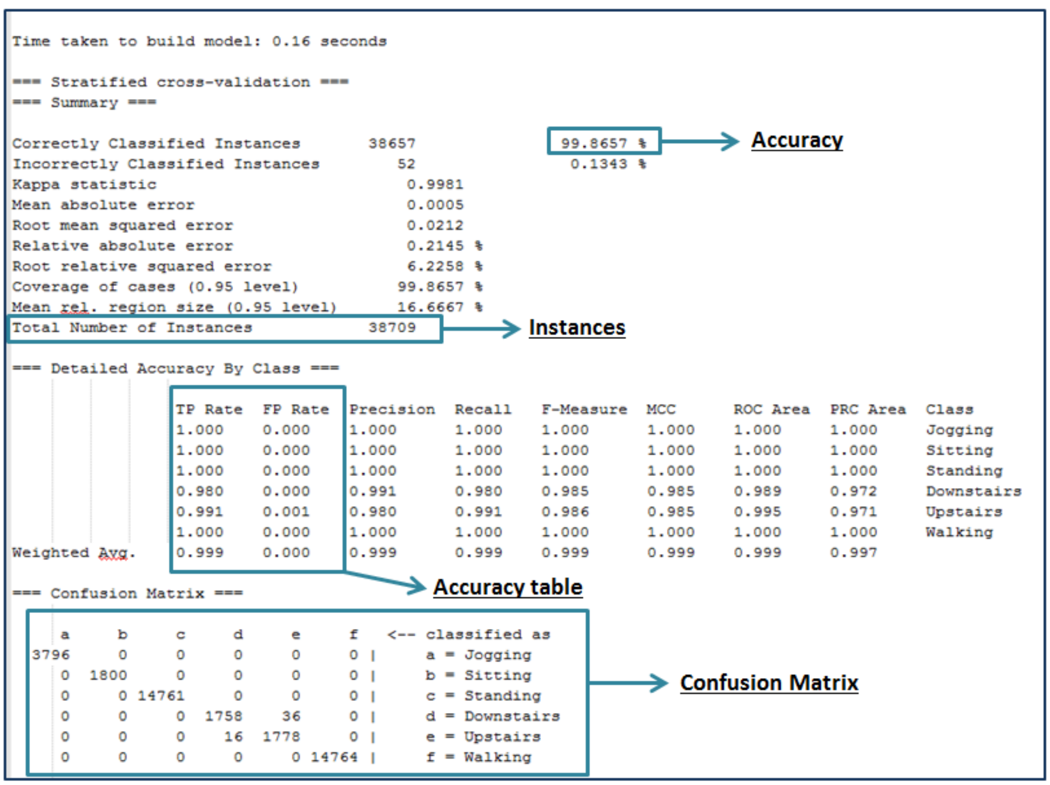
In any case, the process execution time depends on the parameters selected. Some general observations show that when the frequency increases the execution time also increases. On the other hand, as the window size increases the execution time decreases. Finally, as the ratio of window size to step increases the execution time also increases.

The above described process produces a folder named 'ResultsX' where X is a number corresponding to the next number of the already existing 'ResultY' folder. The resulting folder includes all '.csv' and '.mat' files and also the final '.csv' feature file.

5 RESULTS

5.1 Structures outline

Concerning the number of parameters (10) which should be set for any of the following experiments, a structure suitable for visualizing our results was created. The result structures (STRUCTS) consist of two clearly separated parts. The first part is the *'Variables part'* which includes all items that can vary and the change of their values implies the results. The Second part is *'Results part'* where an accuracy table and a confusion matrix table present the effectiveness of each experimental setup. In these tables we have adopted the format of WEKA results, which is the software platform used for the classification process. The true positive column (TPRate), the false positive column (FPRate) and the whole confusion matrix were chosen to simplify the visualization of results. Also, the bottom right cell of accuracy tables represents the percentage of correctly classified instances (accuracy) which is included in the WEKA results. Figure 5.1 shows an example of results coming out from WEKA software.



```
Time taken to build model: 0.16 seconds

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

Correctly Classified Instances        38657              99.8657 %
Incorrectly Classified Instances         52               0.1343 %
Kappa statistic                           0.9981
Mean absolute error                       0.0005
Root mean squared error                   0.0212
Relative absolute error                   0.2145 %
Root relative squared error               6.2258 %
Coverage of cases (0.95 level)           99.8657 %
Mean rel. region size (0.95 level)       16.6667 %
Total Number of Instances             38709

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Jogging
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Sitting
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Standing
                 0.980    0.000    0.991      0.980    0.985      0.985    0.989     0.972     Downstairs
                 0.991    0.001    0.980      0.991    0.986      0.985    0.995     0.971     Upstairs
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Walking
Weighted Avg.    0.999    0.000    0.999      0.999    0.999      0.999    0.999     0.997

=== Confusion Matrix ===

     a     b     c     d     e     f   <-- classified as
  3796     0     0     0     0     0 |     a = Jogging
     0  1800     0     0     0     0 |     b = Sitting
     0     0 14761     0     0     0 |     c = Standing
     0     0     0  1758    36     0 |     d = Downstairs
     0     0     0    16  1778     0 |     e = Upstairs
     0     0     0     0     0 14764 |     f = Walking
```


Figure 5.1: *Example of WEKA results*

5.2 Comparison between reproduced with published results from the WISDM study

An important part of our investigation is the comparison between the published results of WISDM implementation [12] and the results of our reproduced WISDM system. In order to conclude with an optimal system it is necessary to find a respective optimal feature set which includes features either from the MOBIFALL feature set or from the WISDM feature set or both. To do this a reproduction of WISDM study is required. If the results of the reproduction of the WISDM study are approximately the same with the published results, then the reproduction can be used for a comparison of the WISDM feature set and the MOBIFALL feature set performance. The last step might be a comparison of the performances of the two systems while adding or removing features, i.e. finding the optimal feature set.

The setup of our simulation began with choosing the raw data file of the WISDM dataset named 'WISDM_ar_v1.1_raw.txt' as it is available on WISDM website[14]. The window size was set to 200 samples with a step of 200 samples because 200 instances from the raw file (sampling frequency 20 Hz) were needed. The process produced a final '.csv' file with all features described briefly in part 3.3.5 and in [12].

The mentioned raw data '.txt' file includes 36 users which is different from the referred number in WISDM publication (29 users). In order to synchronize the two experiments the final '.csv' was filtered using the filters of WEKA. Specifically, the RemoveWithValues filter was used to remove users 30-36. Then the instances of our experiments were approximately the same. The publication refers to a total of 4526 instances in contrast to our simulation which produced 4184. This difference maybe occurs because while we were processing a subset in the raw file which represented an activity of a user and the number of samples was less than the window size (200) we ignored the samples and moved to the next file. To describe in more detail we split the raw file to smaller ones so each of these represented an activity of a user. Each smaller file was named depending on the activity that is included, the user number (user id) and the trial number of user. If any of these files included less than 200 (window size) samples the file was ignored as signal noise. The above may explain why the number of instances of the two experiments was different.

[14] http://www.cis.fordham.edu/wisdm/dataset.php

Another useful observation considering the instances of the final '.csv' result file is that for some user ids the total instances shown in [12] (Table 1) are different from our number of instances. For example, user id 4 is reported to have 183 'examples' of 43 features. This means that in the raw file there must be at least 183 X 200 samples tagged with user id 4. If we suppose that it is true the samples of user id 4 must be at least 36600. This is not true because the total samples of id 4 are 11371. Probably a change in user ids or a simple mistake in Table 1 might be the reasons for this difference. On the other hand, our result '.csv' includes 54 instances of user id 4 which is the real number of instances that can be extracted from 11371 samples (11371/200 = 56.85 close to 54).

The above instance difference for user id 4 is not the only one. As a result, we state that the way that the instances are extracted in the original WISDM code differs from that of the reproduction. Another reason might be the differences of the whole dataset either in user ids or in the included records.

Although the instances in the above occasions were different, the activity classification results were similar in numbers as can be seen in STRUCTS 22, 23, 24. Table 5.1 summarizes the published and reproduced results giving the percentage of instances correctly predicted. We purposely only include the results for the three classification algorithms (J48, Logistic, Multilayer perceptron), since these were used in the relevant WISDM publication.

***Table 5.1**: Reproduced and published results from the WISDM study*

	Published Results			Reproduced Results		
	J48	Logistic	Multilayer Perceptron	J48	Logistic	Multilayer Perceptron
Walking	89.9	93.6	91.7	90.8	93.8	95.3
Jogging	96.5	98.0	98.3	98.5	98.6	99.0
Upstairs	59.3	27.5	61.5	65.5	53.2	79.3
Downstairs	55.5	12.3	44.3	55.6	49.7	69.4
Sitting	95.7	92.2	95.0	97.0	94.1	94.6
Standing	93.3	87.0	91.9	97.0	94.6	90.4
Overall	85.1	78.1	91.7	88.3	87.5	92.4

5.3 The proposed optimal feature set

One of the objectives of this study is to propose an optimal feature set that will optimize the performance of an implementation of a HAR system on mobile devices. This proposed feature set must lead to better classification accuracies per activity and consequently to a better overall classification accuracy than the published accuracies of the WISDM implementation.

As it is already mentioned in part 5.3, WISDM implementation refers to a dataset that contains 29 users. Therefore, the proposed feature set should be tested with the WISDM dataset after the subtraction of some users (7 users). This synchronization is achieved with a filter included in WEKA’s filter collection named ‘RemoveWithValues’. For reasons of simplicity, the proposed optimal feature set is named PFS. Table 5.2 shows the published results from the WISDM study in contrast with PFS’s results.

***Table 5.2**: PFS’s and published results from the WISDM study*

	Published Results			PFS’s Results		
	J48	Logistic	Multilayer Perceptron	J48	Logistic	Multilayer Perceptron
Walking	89.9	93.6	91.7	99.4	98.3	99.8
Jogging	96.5	98.0	98.3	99.1	99.4	99.6
Upstairs	59.3	27.5	61.5	85.2	79.5	92.5
Downstairs	55.5	12.3	44.3	87.4	77.4	91.5
Sitting	95.7	92.2	95.0	97.0	97.5	98.0
Standing	93.3	87.0	91.9	99.4	97.0	99.4
Overall	85.1	78.1	91.7	96.7	94.9	98.2

The PFS includes all features from the MOBIFALL implementation except from 4 features. The PFS’s features will be described briefly in part 6.1.

As it can be seen in table 5.2, the PFS performs much better than the feature set from the WISDM publication. Moreover, figure 5.2 illustrates a comparison between the overall accuracies for the three classification algorithms used in the WISDM publication and the PFS’s accuracies. All the above results are included in STRUCTS 57, 58, 59 of the ANNEX I.

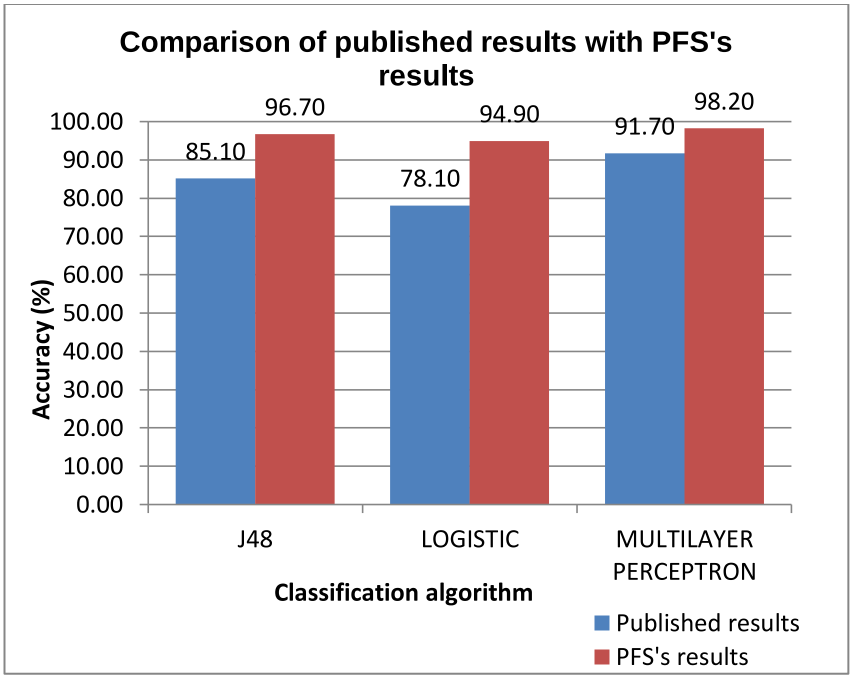


Figure 5.2: *Comparison of published results with PFS's results*

5.4 The quality of the dataset

An additional interesting question we explored in the context of the present work relates to the influence of the dataset's quality on the classification accuracies obtained for each activity.

Table 5.3: *Same number of users (29) in published results using the WISDM dataset and reproduced results using the MOBIFALL dataset.*

	Published Results WISDM data (29 users)			Reproduced Results MOBIFALL data (29 users)		
	J48	Logistic	Multilayer Perceptron	J48	Logistic	Multilayer Perceptron
Walking	89.9	93.6	91.7	96.5	97.1	98.2
Jogging	96.5	98.0	98.3	97.0	99.0	100
Upstairs	59.3	27.5	61.5	76.3	76.3	80.3
Downstairs	55.5	12.3	44.3	73.0	80.5	77.6
Sitting	95.7	92.2	95.0	98.9	97.1	94.8
Standing	93.3	87.0	91.9	99.8	99.5	99.8
Overall	85.1	78.1	91.7	94.7	95.4	95.9

In order to evaluate this influence, if such an influence indeed exists, a comparison of the published results from the WISDM study and the reproduced results using the

MOBIFALL dataset (29 users included), was thought necessary. For this comparison the same three algorithms that were used in the WISDM publication were also tested.

As it can be seen in table 5.3, the way in which a dataset is generated is significant for the classification results. In many datasets, the activities that are made by users are following a 'scenario'. For example, the sitting activity in the MOBIFALL dataset begins with the user in standing position and continues with the user in sitting position. On the other hand, in the WISDM dataset sitting activity begins and ends with the user in sitting position. This can be seen in the plots of section 3.9 where all the signals from all the investigated activities are presented. Such differences in the creation of datasets seem to influence the classification results.

5.5 Overall comparison of results

In this section all interesting results of our investigation will be compared. In the following subsections all parameters will be set as constants except one which will be varied each time. This will give us the significance of the selected variable and the impact of changing its value to the whole experiment.

5.5.1 The impact of changing dataset's size

A common observation in classification process is that the classification algorithms tend to perform differently when there is a changing the size of the dataset. Figure 5.3 confirms this fact by presenting the percentage accuracies of each algorithm used for classification when the MOBIFALL dataset is used. Blue columns show the performance of the algorithms for a smaller (21 subjects less) dataset than the red columns. It is noteworthy that all algorithms perform better with a larger dataset. Also, a remarkable observation is that when the dataset is decreased, the Logistic, Multilayer perceptron and LMT algorithms tend to perform equally well (99.51 % accuracy). All accuracies shown below are also included into STRUCTS 1, 2, 3, 4, 5 (50 users) and 66, 67, 68, 69, 70 (29 users).

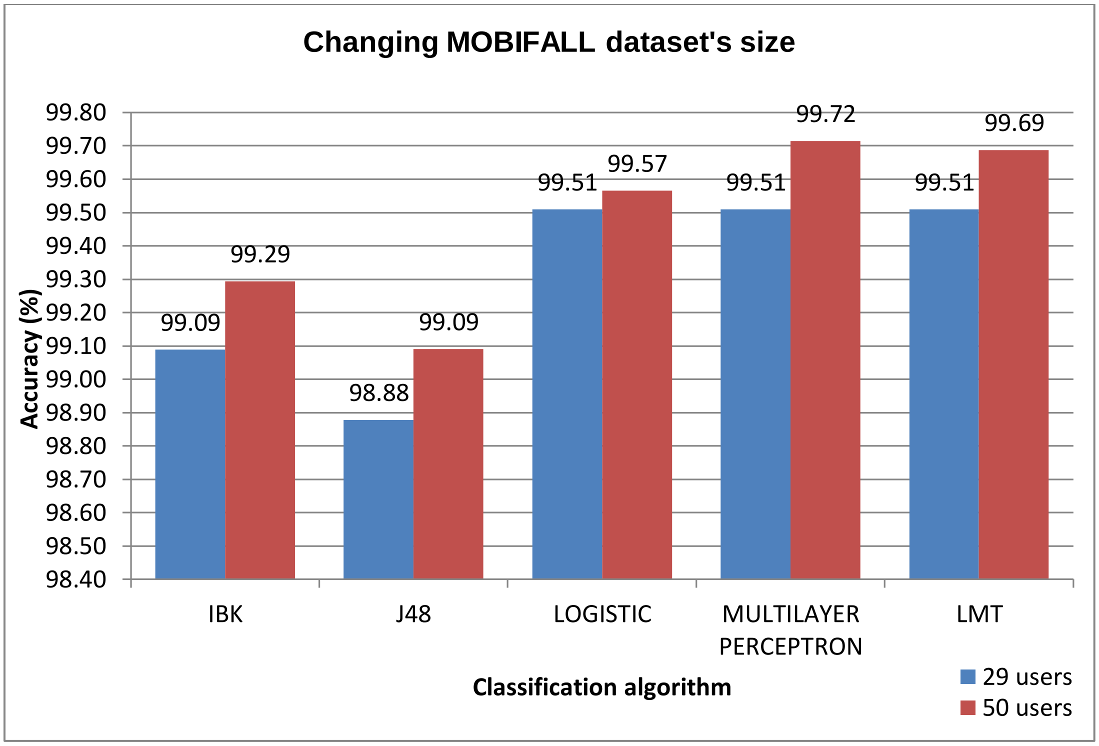


Figure 5.3: *Changing MOBIFALL dataset's size*

A similar comparison for the WISDM dataset is presented below in figure 5.4. Concerning the observations of this column diagram, one could mention that IBK, J48 and LOGISTIC algorithms seem to perform better when the dataset is larger, following the MOBIFALL's implementation behavior. The MULTILAYER PERCEPTON and LMT algorithms perform worse in the same case. This abnormality can be explained if we assume that the added data for the users 2, 4, 19, 20, 25, 32 might include samples that confuse these two algorithms. More specifically, the results might be normalized for the WISDM data if all the users included in the experiment had performed all the activities. Also such an abnormality might be due to the fact that different mobile devices were used for the collection of the WISDM data.

Furthermore, we should also point out that the best performance in classifying the WISDM data is achieved when using IBK nearest neighbor algorithm (1-nearest neighbor). This classification algorithm was not used for the original WISDM implementation [12]. Also when comparing the MOBIFALL implementation accuracies with the WISDM implementation accuracies, shown in figures 5.2, 5.3 respectively, it can be clearly pointed out that the MOBIFALL's setup leads to a better performance. All accuracies shown below are also included into STRUCTS 11, 12, 13, 14, 15 (36 users) and 21, 22, 23, 24, 25 (29 users). These accuracies were achieved by using the same feature set calculated for the WISDM dataset.

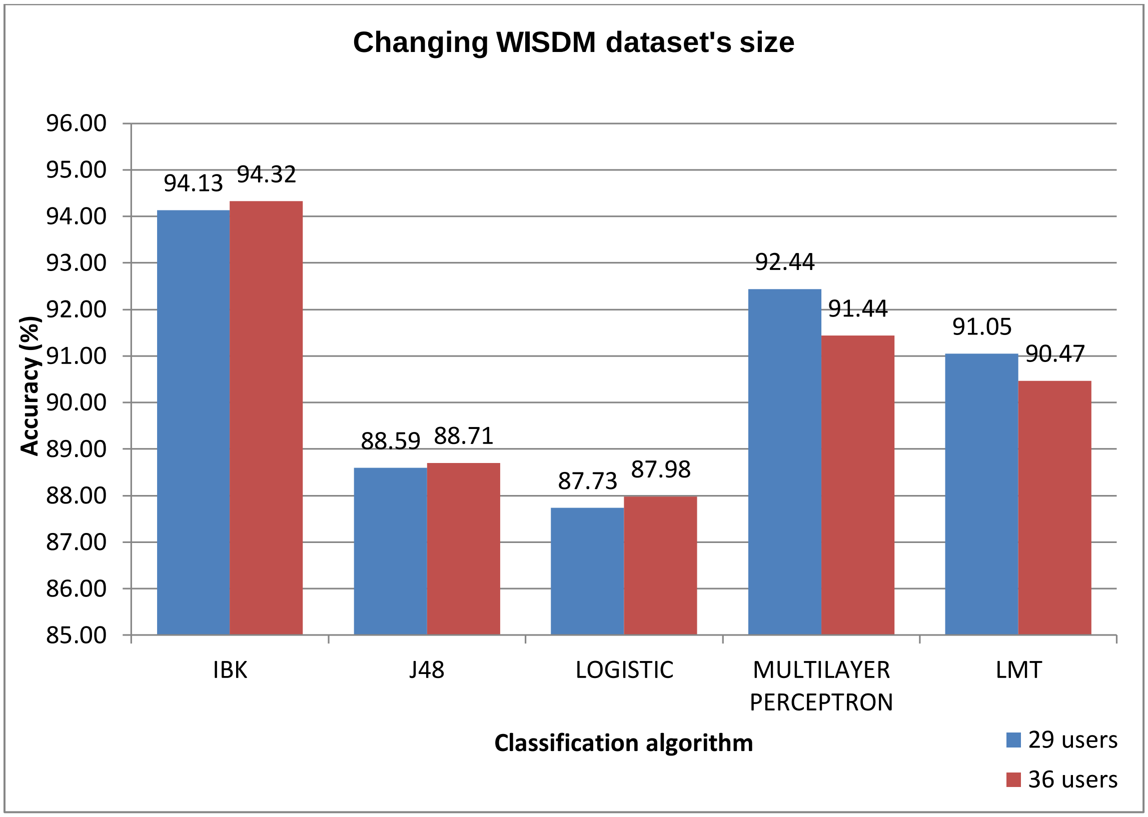


Figure 5.4: *Changing WISDM dataset's size*

5.5.2 The impact of changing window size and step

In the scientific field at hand (human activity recognition), many efforts have already been made in order to maximize the performance. During those efforts many different values for window size and step parameters have been used. *Oresti Banos et al.* [81] summarized these different values of window size used in the literature and also suggest optimal values depending on the activities that are examined. In this publication it is mentioned that, '*For those activities that determine an effective translation of the subject (e.g. walking, jogging, running), it is seen that* W_{min_size} *spans from 0.25 to 0.5 s, while a maximum performance is obtained for* W_{max_perf} *values between one and 1.5 s*'. In conclusion, it is also said, that '(…) *from the results, reduced windows (2 s or less) are demonstrated to provide the most accurate detection performance*'. In addition, regarding the step parameter the most used value in the literature [16], [82], [83] is a percentage of about 50% of the window size.

In order to achieve the best performance in our study a large number of tests were executed. These tests were related to these two key parameters. Although the usage of a window size of 2 seconds (sampling rate = 20Hz), which means 40 samples from the dataset and a step (overlap) of 50% percentage was tested, we didn't obtained the best accuracy. The optimal results were obtained when a window size of 5 seconds (100 samples) and an overlap of 20% were used. These results are presented and discussed in Section 6.3. Figures 5.4, 5.5 show the impact of changing window size and step in the MOBIFALL dataset with MOBIFALL feature set. For this example comparison we used a window size of 40 samples and step of 20 samples (STRUCTS 31, 32, 33, 34, 35) and a window size of 200 samples and a step of 100 samples (STRUCTS 1, 2, 3, 4, 5) to show that the proposed values for these parameters are not always the optimal. Also, a window size of 40 samples and step of 40 samples and window size of 200 samples and step of 200 samples was used for WISDM comparison.

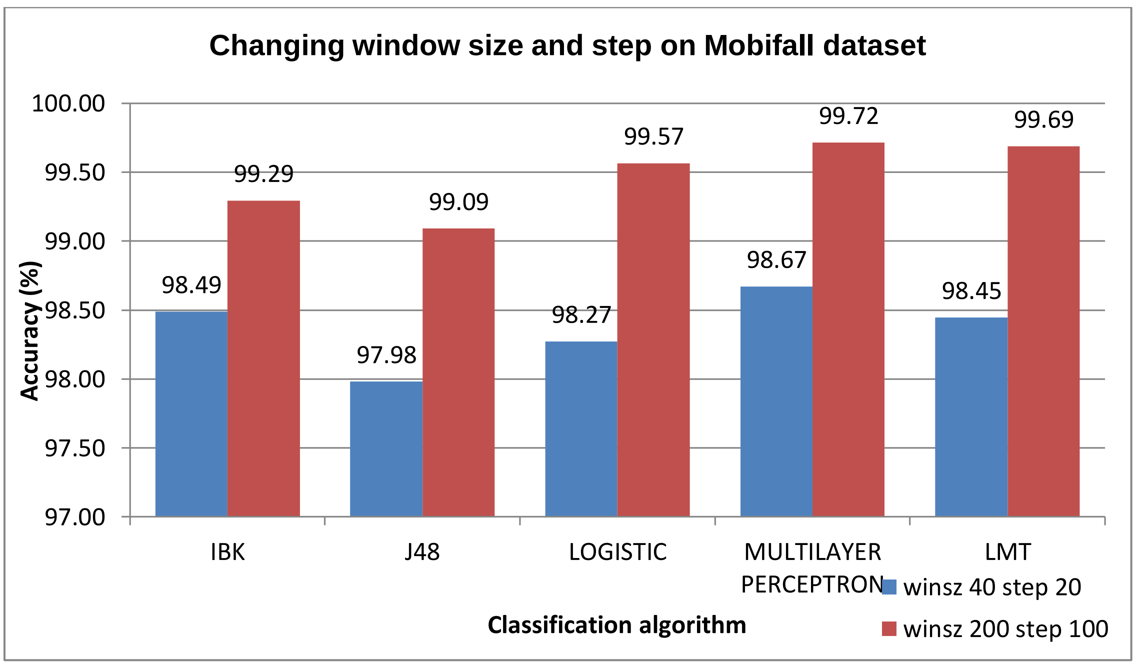


Figure 5.5: *Changing window size and step on MOBIFALL dataset*

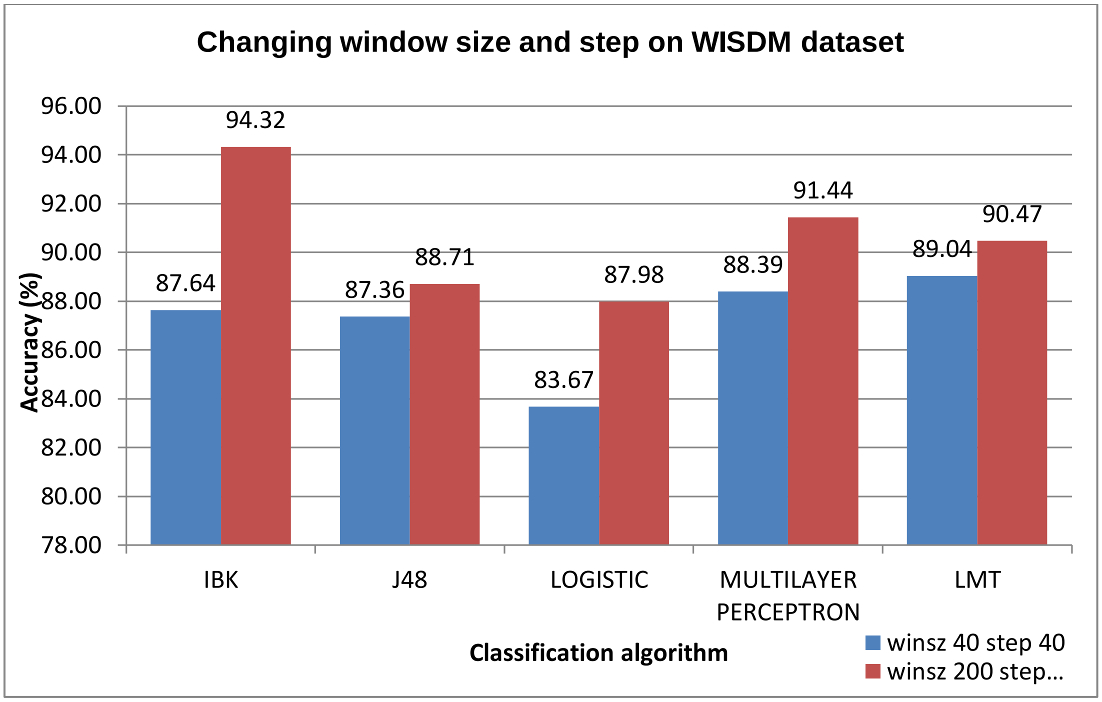


Figure 5.6: *Changing window size and step on WISDM dataset*

5.6 Effect of the position of the smartphone

Considering the weaknesses of human activity recognition systems for ADLs, it is remarkable that climbing stairs (upstairs, downstairs) is not recognized as well as the other activities. As we can see in STRUCTS 1-70, commonly the worst true positive rates appear when classifying upstairs or downstairs activities. In STRUCT 10 a satisfactory true positive rate is achieved for upstairs (0.998) and in STRUCT 6 a satisfactory true positive (0.974) for downstairs. However, in general terms the accuracies of activity recognition for stairs activities are disappointing.

In order to improve the results for these activities, an extra input is needed. An additional parameter in combination with acceleration values for x, y and z axes might result to more accurate recognition. A better observation of acceleration values of y axis (vertical) will help to improve the weakness. To state our opinion we suppose that the ‘normal’ position of a smartphone in trousers pocket is when the bottom of the screen is down to the pocket’s bottom and the top section (usual place of the on/off button) is to the opposite side. On the other hand, ‘abnormal’ position is when the bottom of the screen is up to the pocket’s top and the top section (usual place of the on/off button) is to the opposite side.

Let acc_y be the value of acceleration of y axis when a subject is descending stairs, the smartphone is placed with a ‘normal’ direction and acc_y is a negative number. When the same subject is ascending stairs with the same orientation for mobile device acc_y will be a positive number. As a result if we extract a feature for example ‘the mean value of acceleration for y axes’ when the same subject is descending stairs this will be negative. This is a clue for recognizing such activities with the help of a new input, the position of smartphone into the pocket.

5.7 Classification algorithm’s execution time

This master thesis is directed in producing an optimized implementation of a human activity recognition system for “smart” mobile devices. Considering the small size of such devices and their substantial computing power, the classification algorithm’s execution time has to be optimized.

In our experimental setup, an output that is capable to describe the execution time of each classification algorithm is the time taken to build the classification model from the WEKA’s result buffer. More specifically, WEKA outputs the time taken to build model of the selected classification algorithm in seconds. All the classification algorithm’s execution times that refer to the tests (STRUCTS) included in the ANNEX I are presented in table 5.4.

As it can be seen in the following table, IBK algorithm (K-nearest neighbor classification algorithm, k=1) seems to perform better in all the tests that have been made. Due to the simplicity of this algorithm, the maximum time taken to build its classification model in our tests was 0.21 seconds. This time is proportional to the time that this algorithm needs to be totally completed.

For all stated above, the proposed algorithm for an implementation of a HAR system in a mobile device is IBK.

Table 5.4: *Time in seconds needed to build models of classification algorithms*

				DATASET											
				MOBIFALL						WISDM					
					Classification Algorithm						Classification Algorithm				
Feature Set	S. Rate	Window Size	Step	Users	IBK	J48	LOG	MUL	LMT	Users	IBK	J48	LOG	MUL	LMT
MFS	20	200	100	50	0.03	1.59	45.35	386.94	45.6	36	0.03	2.84	201.94	753.17	656.97
WFS	20	200	200	50	0.03	0.62	9.22	126.29	123.29	36	0.03	1.36	98.9	171.57	122.62
WFS	20	200	200	29	0.02	0.87	9.27	78.23	28.44	29	0	0.94	45.6	65.87	37.69
MFS	20	200	100	40	0.02	1.14	20.23	456.86	20.66	-	-	-	-	-	-
MFS	20	40	20	50	0.14	12.76	317.79	1863.4	662.35	-	-	-	-	-	-
WFS	20	40	40	-	-	-	-	-	-	36	0.03	13.53	2405.96	358.13	2093.66
MFS	20	200	100	29	0.02	0.98	13.98	206.62	12.09	-	-	-	-	-	-
PFS	20	100	20	50	0.12	6.52	64.43	1752.79	789.86	36	0.02	23.68	1809.63	3399.93	1513.39
PFS	40	100	20	50	0.21	24.27	203.91	1956.99	1068.7	-	-	-	-	-	-
PFS	20	200	200	-	-	-	-	-	-	29	0.01	0.83	60.19	288.43	100.33

6 PROPOSED SYSTEM

As it is already mentioned in chapter 3.7, our study's objective is to create an optimal system which recognizes activities of human beings accurately. This will be the result of the comparison of two datasets (MOBIFALL and WISDM). The features which will be included in this hybrid system will be the union of all features included in the MOBIFALL feature set and all the features included in the WISDM feature set (some of them are in common).

6.1 Optimal features

As a result of this investigation a set of features that tend to perform better than all the other feature sets was extracted. These features came out from many tests, through a trial and error process, that have been made either with the MOBIFALL dataset or with the WISDM dataset. The selected optimal features are all the features included in implementation of fall detection with the MOBIFALL dataset [16] except from 4 features which seem to affect the accuracy of human activity recognition and were therefore initially removed. The excluded features are:

- kurtosis of acceleration values in axis x
- kurtosis of acceleration values in axis y
- kurtosis of acceleration values in axis z
- and the spectral centroid.

The resulting optimal feature set includes 65 individual features. These include all the features described in chapter 3.4.6. For all these features, absolute values are also computed. The idea of adding the absolute values as separate features came from the original MOBIFALL implementation where such absolute values might help to account for different mobile phone orientation into the pocket. It is remarkable that absolute values of kurtosis in all respective axes do not affect the accuracy of our experiment negatively. In contrast they improve the performance of classification and they are included in the final optimal feature set. Moreover, spectral centroid is the key feature which seems to affect negatively the results concerning the activities including upstairs and downstairs. For these activities, the worst accuracies in human activity recognition were achieved regarding all the tests that have been made. This might be a point for future exploration. If the accuracies of the latter two activities were improved, better overall results will come out of our experiments. As it can be seen in the following STRUCTS the algorithm which performs better (best accuracy, execution time) in such activities (climbing stairs) is the algorithm that meets the circumstances to be the optimal algorithm for an implementation of HAR in a mobile device. These activities tend to affect negatively the accuracy of all algorithms.

6.2 Proposed system results

In ANNEX I we present the proposed system's STRUCTS (41 - 50). All these tests were performed with a sampling rate of 20Hz, a window size of 100 samples (optimal), and a step of 20 samples (optimal). These values are the optimal as it comes out of the comparison of the accuracies in all the tests made during this research. STRUCTS 41 – 45 present the results when the MOBIFALL data set is used and STRUCTS 46 – 50 present he results with the WISDM dataset. The best overall accuracy is achieved in STRUCT 41, in which the MOBIFALL dataset is used in combination with the optimal feature set mentioned in section 6.1.

To filter the best accuracies achieved in our study we can select STRUCTS 41-50. STRUCTS 41-45 show PFS's results when the MOBIFALL dataset is used. STRUCTS 46-50 show PFS's results when the WISDM dataset is used. As it can be seen in figure 6.1, the best performance is achieved when the MOBIFALL dataset and the PFS are used while the classification algorithm is IBK. This algorithm is included in WEKA's algorithm set and represents *k-nearest neighbor algorithm* for classification. As it has already been mentioned, in each test made with IBK algorithm the parameter *k* was set to 1. In the second position of the ranking comes multilayer perceptron. This algorithm included in WEKA uses backpropagation to classify instances. It is a network that can either be built by hand, or created by an algorithm or both. Although it is a very accurate algorithm, it takes a lot of time to complete. The execution time of each algorithm will also be inspected briefly in this thesis. On the other hand, IBK with *k=1* performs slightly better. If the execution time of IBK with multilayer perceptron is compared, IBK performs better in aliquot time. Remarkable are also the results of LMT algorithm which was an empirical selection in our study. This algorithm came third in ranking but still its execution time is not competitive.

In this chapter we present the results of the proposed system using MOBIFALL data (STRUCTS 51 - 55). These tests were performed in order to define the impact of changing the sampling rate while the other parameters are steady.

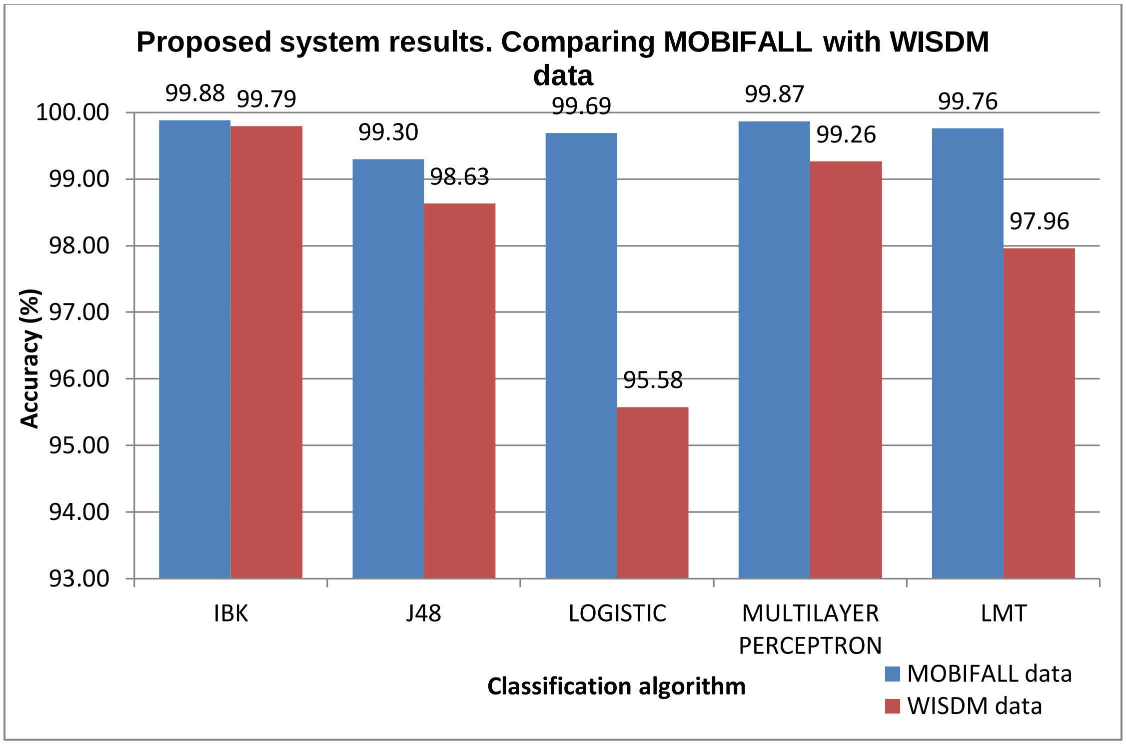


Figure 6.1: *Proposed system results. Comparing MOBIFALL with WISDM data*

6.3 The impact of doubling the sampling rate

After the completion and evaluation of results of tests, presented in ANNEX I, an additional question arose. The question relates to the sampling rate used and can be stated as follows: 'What is the effect on the results from a change in the sampling rate used to acquire the data from the smartphone? In order to answer this question the MOBIFALL data were used in combination with PFS's features. Window size was set to 100 and step to 20. These values are the proposed optimal setting as it comes out of the comparison of the accuracies in all the tests made during this research.

As it is shown in figure 6.2, when the sampling rate is doubled, from 20 to 40 Hz, the results deteriorate. For the purposes of this thesis, all the experiments were based on the use of 20 Hz sampling rate. This sampling frequency was selected as it allowed for the comparative analysis of results with the published results of the WISDM study. In addition, the use of such a low sampling frequency leads to an energy efficient implementation on a mobile device which is a future target for this study. Obviously, if a different sampling frequency must be used; different values for the window size and the overlap parameters must be used respectively in order to achieve the best classification accuracies.

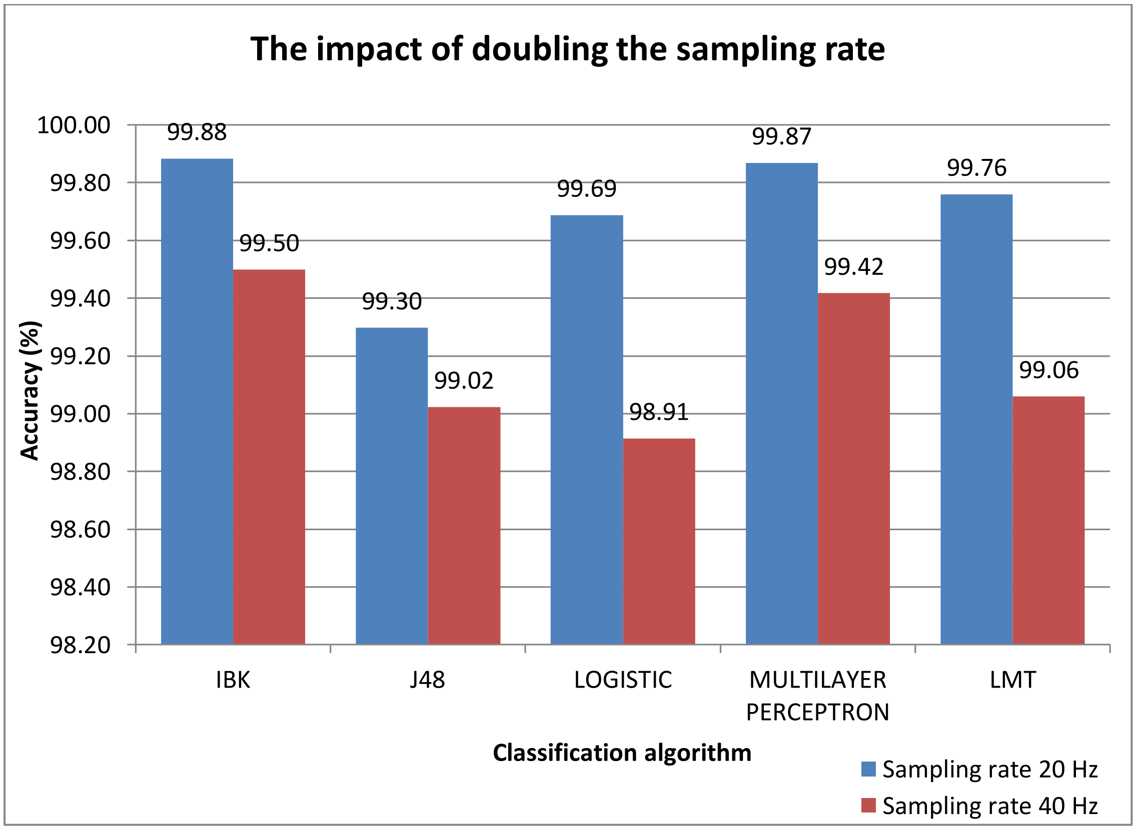


Figure 6.2: *Comparing results of different sampling rates in MOBIFALL data using PFS's features*

In addition, it should be noted that, as shown in figure 6.2, the results obtained with doubled sampling rate and the same values for the other parameters (window size of 100 samples, overlap of 20%), are of lower accuracy. This is an unexpected result that needs to be further investigated in the future. The final conclusion of our tests with both datasets is that the optimal setup for achieving best results (accuracies %) is with window size 100, step 20 and sampling rate 20 Hz.

7 CONCLUSIONS

This study was set out to explore the concept of human activity recognition (HAR) via the enhanced sensing ability of smartphones. Although such mobile devices involve several sensors, our work on activity recognition concentrates on recognize many of the physical activities that a smart phone user is performing (walking, jogging, sitting, etc.) based on the user's movements, as measured by the smart phone's tri-axial accelerometer.

Many approaches of HAR have been proposed in the literature over the past decade. Vision-based approaches, wearable sensor-based approaches, smartphone-based approaches and combinations of them are the categories which cover the implemented HAR systems. The most innovative approaches of detecting and classifying activities of daily life are based on the abilities of smartphone devices. Triggered by those approaches, this study proposes a HAR system based on data collected from the acceleration sensor of smartphone devices.

The HAR system presented recognizes accurately six activities of daily living (walking, jogging, upstairs, downstairs, sitting and standing). For the purposes of this comparative investigation were used two datasets: *MOBIFALL and WISDM.* These two datasets were synchronized in order to be compared. The MOBIFALL data were interpolated and the sampling rate was downgraded to 20 Hz. This was the sampling rate used in the publication of the WISDM dataset. When the two datasets were synchronized and unified to a simple file format they were tested with the same parameters. Finally, an optimal setup is proposed for a future implementation of HAR system in mobile devices.

7.1 Discussion

In the current study, the need of comparing different implementations of HAR was faced. After the selection of the datasets to be compared and the comparison process, useful observations and conclusions were extracted.

The MOBIFALL implementation for detecting everyday falls was available to us. As a result there was a need to convert the MOBIFALL implementation to an approach that uses only acceleration data (gyroscope data processing was removed) so that it can be compared with the WISDM implementation. Also, the MOBIFALL data were interpolated in order to downgrade the data sampling rate to 20Hz similarly with the WISDM approach. On the other hand, the WISDM implementation was not available to us. Consequently a reproduction of WISDM study was required.

The reproduction of the WISDM study was achieved following the instructions of features implementation included in the latest publication relative to the WISDM project. After this procedure, a comparison of the published results of the WISDM study with the reproduced results was made. The results that came from the reproduction of the WISDM implementation were similar to the published. More specifically, in the most of the cases the reproduced results differed slightly from the published results. In the cases that the compared results differed a lot from the reproduced, the results of the reproduced study were better. Therefore, the reproduced WISDM implementation was suitable for further analysis and for deciding whether the features used in the WISDM implementation should be contained in the proposed system’s features.

The reproduction of the WISDM implementation and the comparison of its results with the published results of the WISDM project led to the proposed feature set (PFS). PFS includes all the features contained in the MOBIFALL implementation except from kurtosis features for the three axes respectively and the spectral centroid feature. These four features seem to deteriorate the results of the classification process and they were removed. Thus, the used PFS (65 features) is the proposed feature set that is suitable for using in an implementation of a HAR system in mobile devices.

The next investigation point was the significance of the quality of the dataset for the current human activity recognition study. Using a part of MOBIFALL dataset (29 users) and the reproduced implementation of the WISDM study we achieved better classification accuracies than the published accuracies of the WISDM implementation. Therefore, the data collection method and the “scenarios” that can probably be seen in the activities contained in the used datasets, have a great significance for the classification process and consequently for the activity recognition.

During the effort made to optimize the classification accuracies, many parameters were tested with different values. One of them was the dataset’s size. In our work, the two datasets used were tested with their complete number of users (all participants) but also with a part of them. For the purposes of the comparison in both of the two datasets tests have been made with 29 users. In tests made with the MOBIFALL dataset no abnormality was encountered. All classification algorithms seem to perform better when we use a larger dataset. On the other hand, in the test made with the WISDM dataset we faced an abnormality. However the classification algorithms IBK, J48 and LOGISTIC seem to perform better when the dataset is larger, when LMT or MULTILAYER PERCEPTRON is used the classification results are worse when the dataset's size is increased. Such an abnormality can be explained from the fact that the WISDM dataset includes 29 users which did not perform all the activities. Another possible reason for such an abnormality is the different mobile devices that were used during the collection process of the WISDM dataset. As a general conclusion, when changing the dataset's size we observe that the larger the dataset is the better the classification accuracies become. This expected finding, points towards the need for really large reference datasets so that alternative approaches and algorithms will be objectively evaluated.

The testing part of this investigation was a trial and error procedure. Many efforts have been made for all the variables which might affect the classification results. In specific, the selected window size and overlap were parameters that were examined. In the recent relevant literature [81], the proposed values for these parameters were 2 seconds for the window size and 50% for the overlap. Although these values have been tested, the optimal classification accuracies were obtained with a 5 seconds window size and a 20% overlap. As a result, the optimal window size and overlap depend on the dataset used in human activity recognition. Another observation is that as the window size is reduced the execution time is increased. In addition, when the overlap percentage is closer to 100% the execution time is reduced. Indicative tests that have been made for the parameters window size and overlap are included in the ANNEX I.

In all the tests that are presented in the ANEX (STRUCTS 1-70), the classification accuracies achieved for the activities walking upstairs and walking downstairs were not as well as the accuracies of the rest four activities (walking, jogging, sitting, standing). This is happening because the activities of climbing stairs tend to confuse the classification algorithms. What is remarkable is the fact that in most of the confusion matrixes included in the ANNEX I the "*upstairs*" activity tends to be confused with the *"downstairs"* activity and vice versa. This is the main weakness of our investigation. An extra input might be used for improving the recognition accuracies of these activities.

As it is proposed in this study, the position (direction) of the smartphone into the trousers pocket may be the extra input that would lead to better classification accuracies. As it can be seen in the plotted signals of these two activities, when a participant begins to walk down the stairs, a negative acceleration is created along the y axis if the smartphone is placed into the pocket normally. This is explained briefly in part 5.6. Respectively, the acceleration along the y axis when the participant begins to walk up the stairs should be positive. This is the clue that can be used for separating the two commonly confused activities.

As an optimization procedure in this investigation we found and we proposed the classification algorithm that performs better than the others that have been tested in terms of execution time. The proposed algorithm for an implementation of a HAR system in a mobile device is IBK (K-nearest neighbor classification algorithm, k=1) because of its simplicity and its good performance in all the tests that have been made. Concerning the best accuracies that have been achieved in the classification process (STRUCT 41), the time was consumed to build the model of the IBK classification algorithm was 0.12 seconds according to the result buffer of the WEKA.

The effect of an additional parameter was examined in this comparative study. We specifically explored the influence of the sampling frequency or sampling rate and its impact on accuracy. As it can be seen in all the tests made (ANNEX I), the different sampling rates used were 20 Hz and 40 Hz. The sampling frequency of 20 samples per second was used to enable the comparison between the MODIFALL dataset implementation and the WISDM dataset implementation. Also, such a low sampling frequency was tested because in a future implementation of the proposed system in a mobile device, the lower the sampling rate used the more energy efficient the system will be.

However, to discover the impact of changing the sampling frequency while the other parameters are steady, tests were made with the use of 40 Hz sampling rate. These tests proved that depending on the sampling frequency which is used in a test, the other parameters must have different values in order to achieve the best classification accuracies.

At the end of the day, the best classification accuracies achieved when the MOBIFALL dataset is used in conjunction with the PFS's features. Concerning the different classification algorithms that were tested, the IBK algorithm performed better than the others. In addition, the time that had been consumed to build the model of this algorithm was the lowest. As it is already discussed in part 6.2, IBK achieved 99.88% overall accuracy when the MOBIFALL dataset (50 users) was used. Also the overall activity recognition accuracy was 99.77% when the WISDM dataset (36 users) was used. These overall accuracies were achieved with a sampling rate of 20 Hz, a window size of 100 samples and an overlap of 20%. As a result, the proposed algorithm for an implementation of a human activity recognition system in a mobile device is IBK.

In this study, a machine learning approach is proposed. Machine learning usually produces more accurate and reliable results, while threshold-based algorithms are faster and simpler. Human activity recognition systems are responsible for giving crucial information about humans and their physical activities. The information extracted by HAR systems is used to reach to significant decisions for humans especially when the system monitors a patient or an elder person. Thus, the quality and the specificity of information are necessary.

7.2 Directions for future work

This master thesis proposed a human activity recognition system that can be implemented in a mobile device and recognize six activities of daily living accurately. Although the results of the recognition process were satisfactory, there are several lines of research arising from this work which should be pursued and directions that should be followed.

One such direction should be to investigate whether the use of gyroscope data that are available, might lead to better human activity recognition results. This could happen by using the gyroscope data for defining the position or direction of the smartphone into pocket. This is an input to the system that will improve the performance of classification algorithms and thus optimize the classification results for ADL's like walk-up the stairs and walk-down the stairs.

All above must be examined under the concept of implementing the proposed system in a mobile device. This implies that such a system must be energy efficient considering the small size of such devices and their substantial computing power. After such an examination the proposed system must be implemented in a mobile device.

This study is an approach of recognizing the ADLs with the use of smartphones. Despite their significance, the ADLs do not measure the full range of activities necessary for independent living in the community. To partly fill this gap it is necessary to examine human activity recognition from the aspect of "instrumental activities of daily living," or IADLs. The ultimate goal of this approach is to model the instrumental ADLs (IADLs) as a set of elementary ADLs and thus develop methods to detect and measure critical “behaviors”, i.e. extend our ability to monitor events and activities towards detecting complex behaviors.

REFERENCES

[1] Commonwealth of Australia, "Department of Health and Ageing. Ageing and aged care in Australia.," 2008. [Online]. Available: http://www.health.gov.au.

[2] Connell, C. Grealy, K. Olver and J. Power, "Urbis for the Department of Health and Ageing," 2008. [Online]. Available: http://www.policypointers.org.

[3] D. Tennenhouse, "Proactive computing," *Communications of the ACM,* vol. 43, no. 5, pp. 43-50, May 2000.

[4] S. Consolvo, P. Roessler, B. E. Shelton and L. Anthony, "Technology for care networks of elders," *Pervasive Computing, IEEE,* vol. 3, no. 2, pp. 22-29, 2004.

[5] T. B. Moeslund, A. Hilton and V. Kruger, "Computer Vision and Image Understanding," *Elsevier,* vol. 104, no. 2-3, pp. 90-126, 2 August 2006.

[6] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola and I. Korhonen, "Activity classification using realistic data from wearable sensors," *Information Technology in Biomedicine, IEEE Transactions,* vol. 10, no. 1, pp. 119-128, January 2006.

[7] N. H. Lovell, W. Ning, E. Ambikairajah and B. G. Celler, "Accelerometry Based Classification of Walking Patterns Using Time-frequency Analysis," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, Lyon, 2007.

[8] Y. Tao, H. Hu and H. Zhou, "Integration of Vision and Inertial Sensors for 3D Arm Motion Tracking in Home-based Rehabilitation," *The International Journal of Robotics Research,* vol. 26, no. 6, pp. 607-624, 2007.

[9] H. Leutheuser, D. Schuldhaus and B. M. Eskofier, "Hierarchical, Multi-Sensor Based Classification of Daily Life Activities: Comparison with State-of-the-Art Algorithms Using a Benchmark Dataset," *PLoS ONE,* vol. 8, no. 10, 2013.

[10] Z. Mi and A. A. Sawchuk, "Human Daily Activity Recognition With Sparse Representation Using Wearable Sensors," *Biomedical and Health Informatics, IEEE Journal,* vol. 17, no. 3, pp. 553-560, 2013.

[11] N. Győrbíró, Á. Fábián and G. Hományi, "An Activity Recognition System For Mobile Phones," *Mobile Networks and Applications,* vol. 14, no. 1, pp. 82-91, 2009.

[12] J. R. Kwapisz, G. M. Weiss and S. A. Moorel, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter,* vol. 12, no. 2, pp. 74-82, 2011.

[13] P. Siirtola and J. Röning, "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 1, no. 5, pp. 38-45, 2012.

[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," in *Future Information Technology (FutureTech), 2010 5th International Conference*, Busan, 2010.

[15] Y.-S. Lee and S.-B. Cho, "Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer," in *Hybrid Artificial Intelligent Systems*, Springer Berlin Heidelberg, 2011, pp. 460-467.

[16] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. G. Spanakis and M. Tsiknakis, "The MobiFall Dataset: Fall Detection and Classification with a Smartphone," *International Journal of Monitoring and Surveillance Technologies Research (IJMSTR),* vol. 2, no. 1, p. 13, 2014.

[17] J. M. Warren, U. Ekelund, H. Besson, A. Mezzani, N. Geladas and L. Vanhees, "Assessment of physical activity – a review of methodologies with reference to epidemiological research: a report of the exercise physiology section of the European Association of Cardiovascular Prevention and Rehabilitation," *European Journal of Preventive Cardiology,* vol. 17, no. 10, pp. 127-139, 2010.

[18] R. J. Shephard, "Limits to the measurement of habitual physical activity by questionnaires," *British journal of sports medicine,* vol. 37, no. 3, pp. 197-206, 2003.

[19] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR),* vol. 43, no. 3, 2011.

[20] C. Schuldt, I. Laptev and B. Caputo, "Recognizing human actions: a local SVM approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference*, 2004.

[21] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, "Actions as Space-Time Shapes," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference*, Beijing, 2005.

[22] D. Weinland, R. Ronfard and E. Boyer, "Computer Vision and Image Understanding," *ELSEVIER,* vol. 104, no. 2-3, pp. 249-257, 2006.

[23] M. D. Rodriguez, J. Ahmed and M. Shah, "Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference*, Anchorage, AK, 2008.

[24] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, "Learning Realistic Human Actions from Movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference*, Anchorage, AK, 2008.

[25] Y. Ke, R. Sukthankar and M. Hebert, "Event Detection in Crowded Videos," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference*, Rio de Janeiro, 2007.

[26] N. Ikizler, R. G. Cinbis, S. Pehlivan and P. Duygulu, "Recognizing Actions from Still Images," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference*, Tampa, FL, 2008.

[27] X. Peng, C. Zou, Y. Qiao and Q. Peng, "Action Recognition with Stacked Fisher

Vectors," in *Computer Vision – ECCV 2014*, Zurich, Springer International Publishing, 2014, pp. 581-595.

[28] X. Yang and Y. Tian, "EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference*, Providence, RI, 2012.

[29] J. C. Niebles and L. F. Fei, "A Hierarchical Model of Shape and Appearance for Human Action Classification," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference*, Minneapolis, MN, 2007.

[30] M. Ahmad and S. W. Lee, "Human action recognition using shape and CLG-motion flow from multi-view image sequences," *ELSEVIER,* vol. 41, no. 7, p. 2237–2252, 2008.

[31] Q. Shi, L. Wang, L. Cheng and A. Smola, "Discriminative human action segmentation and recognition using semi-Markov model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference*, Anchorage, AK, 2008.

[32] I. Junejo, E. Dexter, I. Laptev and P. Perez, "Cross-View Action Recognition from Temporal Self-similarities," in *Computer Vision – ECCV 2008*, Marseille, Springer Berlin Heidelberg, 2008, pp. 293-306.

[33] A. Farhadi, M. K. Tabrizi, I. Endres and D. Forsyt, "A latent model of discriminative aspect," in *Computer Vision, 2009 IEEE 12th International Conference*, Kyoto, 2009.

[34] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris and S. M. Ayer, "SHIMMER – A Wireless Sensor Platform for Noninvasive Biomedical Research," *Sensors Journal, IEEE,* vol. 10, no. 9, pp. 1527 - 1534, 2010.

[35] F. Albinali, S. S. Intille, W. Haskell and M. Rosenberger, "Using wearable activity type detection to improve physical activity energy expenditure estimation," in *12th ACM international conference on Ubiquitous computing*, New York, 2010.

[36] J. W. Lockhart, T. Pulickal and G. M. Weiss, "Applications of mobile activity recognition," in *UbiComp '12 Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, Pittsburgh, USA, 2012.

[37] A. Mannini and A. M. Sabatini, "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers," *SENSORS,* vol. 10, no. 2, pp. 1154-1175, 2010.

[38] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *Communications Surveys & Tutorials, IEEE,* vol. 15, no. 3, pp. 1192 - 1209, 2012.

[39] A. BULLING, U. BLANKE and B. SCHIELE, "A Tutorial on Human Activity Recognition Using Body-worn Inertial," *ACM Computing Surveys (CSUR),* vol. 46, no. 3, 2014.

[40] M. Ermes, J. Parkka, J. Mantyjarvi and I. Korhonen, "Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions," *Information Technology in Biomedicine, IEEE Transactions,* vol. 12, no. 1, pp. 20 - 26, 2008.

[41] M. Keally, G. Zhou, G. Xing, J. Wu and A. Pyles, "PBN: towards practical activity recognition using smartphone-based body sensor networks," in *SenSys '11 Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, 2011.

[42] U. Maurer, A. Smailagic, D. P. Siewiorek and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop*, Cambridge, MA, 2006.

[43] Ó. D. Lara, A. J. Pérez, M. A. Labrador and J. D. Posada, "Centinela - A human activity recognition system based on acceleration and vital sign data," *ELSEVIER,* vol. 8, no. 5, p. 717–729, 2012.

[44] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *International Multimedia Conference*, 2009.

[45] T. Brezmes, J. L. Gorricho and J. Cotrina, "Activity Recognition from Accelerometer Data on a Mobile Phone," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, Salamanca, Springer Berlin Heidelberg, 2009, pp. 796-799.

[46] G. Hache, E. D. Lemaire and N. Baddour, "Mobility change-of-state detection using a smartphone-based approach," in *Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop*, Ottawa, 2010.

[47] S. Zhang, P. J. McCullagh, C. D. Nugent and H. Zheng, "Activity Monitoring Using a Smart Phone's Accelerometer with Hierarchical Classification," in *Intelligent Environments (IE), 2010 Sixth International Conference*, Kuala Lumpur, 2010.

[48] M. Liu, "A Study of Mobile Sensing Using Smartphones," *International Journal of Distributed Sensor Networks,* vol. 2013, p. 11, 2013.

[49] W. Z. Khan, Y. Xiang, M. Y. Aalsalem and Q. Arshad, "Mobile Phone Sensing Systems: A Survey," *Communications Surveys & Tutorials, IEEE,* vol. 15, no. 1, pp. 402 - 427, 2012.

[50] A. T. Campbell, T. Choudhury, S. Hu, H. Lu, M. K. Mukerjee, M. Rabbi and R. D. S. Raizada, "NeuroPhone: brain-mobile phone interface using a wireless EEG headset," in *MobiHeld '10 Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handheld*, 2010.

[51] E. Miluzzo, T. Wang and A. T. Campbell, "EyePhone: activating mobile phones with your eyes," in *MobiHeld '10 Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, 2010.

[52] H. Lu, W. Pan, N. D. Lane, T. Choudhury and A. T. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," in

MobiSys '09 Proceedings of the 7th international conference on Mobile systems, applications, and services, 2009.

[53] J. Hicks, R. Nithya, F. Hossein, L. Brent, P. Kannan, R. Mohammad, K. Donnie, S. Joshua, J. John and D. Estrin, "AndWellness: an open mobile system for activity and experience sampling," in *WH '10 Wireless Health 2010*, 2010.

[54] K. Sha, G. Zhan, W. Shi, M. Lumley, C. Wiholm and B. Arnetz, "SPA: a smart phone assisted chronic illness self-management system with participatory sensing," in *HealthNet '08 Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, 2008.

[55] T. Denning, A. Andrew, R. Chaudhri, C. Hartung, J. Lester, G. Borriello and G. Duncan, "BALANCE: towards a usable pervasive wellness application with accurate activity inference," in *HotMobile '09 Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, 2009.

[56] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith1 and J. A. Landay, "Activity sensing in the wild: a field trial of ubifit garden," in *CHI '08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008.

[57] P. Järvinen, T. H. Järvinen, L. Lahteenmaki and C. Sodergard, "HyperFit: Hybrid media in personal nutrition and exercise management," in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference*, Tampere, 2008.

[58] C. Liao, Q. Liu, B. Liew and L. Wilcox, "Pacer: fine-grained interactive paper via camera-touch hybrid gestures on a cell phone," in *CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010.

[59] X. Chen, C. Ho, E. Lim and T. Kyaw, "Cellular phone based online ECG processing for ambulatory and continuous detection," in *Computers in Cardiology*, Durham, 2007.

[60] E. Miluzzo, N. D. Lane, S. B. Eisenman and A. T. Campbell, "CenceMe – Injecting Sensing Presence into Social Networking Applications," in *Smart Sensing and Context*, Springer Berlin Heidelberg, 2007, pp. 1-28.

[61] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin and M. Hansen, "Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype," in *EmNets '07 Proceedings of the 4th workshop on Embedded networked sensors*, 2007.

[62] R. d. Oliveira and N. Oliver, "TripleBeat: enhancing exercise performance with persuasion," in *MobileHCI '08 Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 2008.

[63] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *MobiSys '09 Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009.

[64] X. Bao and R. R. Choudhury, "MoVi: mobile phone based video highlights via collaborative sensing," in *MobiSys '10 Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010.

[65] R. K. Ranayz, C. T. Chouy, S. S. Kanhere, N. Bulusu and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *IPSN '10 Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010.

[66] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox and A. Schmidt, "Micro-Blog: sharing and querying content through mobile phones and social participation," in *MobiSys '08 Proceedings of the 6th international conference on Mobile systems, applications, and services*, 2008.

[67] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo and J. Eriksson, "VTrack: accurate, energy-aware road traffic delay

estimation using mobile phones," in *SenSys '09 Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 2009.

[68] P. Mohan, V. N. Padmanabhan and R. Ramjee, "TrafficSense: Rich Monitoring of Road and Traffic Conditions using Mobile," Microsoft Research India, Redmond, 2008.

[69] L. Deng and L. P. Cox, "LiveCompare: grocery bargain hunting through participatory sensing," in *HotMobile '09 Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, 2009.

[70] S. Sehgal, S. S. Kanhere and C. T. Chou, "http://www.researchgate.net/," 2008. [Online]. Available: http://www.researchgate.net/profile/Salil_Kanhere/publication/229013654_MobiShop_Using_Mobile_Phones_for_Sharing_Consumer_Pricing_Information/links/09e415087b06539ee8000000.pdf?disableCoverPage=true. [Accessed 23 06 2015].

[71] M. Bilandzic, M. Banholzer, D. Peev, V. Georgiev, F. B. Fernandez and A. D. Luca, "Laermometer: a mobile noise mapping application," in *NordiCHI '08 Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, 2008.

[72] E. Kanjo, S. Benford, M. Paxton, A. Chamberlain, D. S. Fraser, D. Woodgate, D. Crellin and A. Woolard, "MobGeoSen: facilitating personal geosensor data collection and visualization using mobile phones," *Personal and Ubiquitous Computing,* vol. 12, no. 8, pp. 599-607, 2008.

[73] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee and A. Sharma, "PRISM: platform for remote sensing using smartphones," in *MobiSys '10 Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010.

[74] J. Dai, X. Bai, Z. Yang, Z. Shen and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference*, Mannheim, 2010.

[75] F. Sposaro and G. Tyson, "iFall: An android application for fall monitoring and response," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, Minneapolis, 2009.

[76] N. Oliver and F. F. Mangas, "HealthGear: Automatic Sleep Apnea Detection and Monitoring with a Mobile Phone," *Journal of Communications,* vol. 2, no. 2, pp. 1-9, 2007.

[77] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth and A. Aucinas, "EmotionSense: a mobile phones based adaptive platform for experimental social psychology research," in *UbiComp '10 Proceedings of the 12th ACM international conference on Ubiquitous computing*, 2010.

[78] E. Miluzzoy, C. T. Corneliusy, A. Ramaswamyy, T. Choudhuryy, Z. Liux and A. T. Campbelly, "Darwin phones: the evolution of sensing and inference on mobile phones," in *MobiSys '10 Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010.

[79] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada and R. Han, "WhozThat? evolving an ecosystem for context-aware mobile social networks," *Network, IEEE,* vol. 22, no. 4, pp. 50 - 55, 2008.

[80] D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. R. Ortiz, "Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine," in *Ambient Assisted Living and Home Care*, Vitoria-Gasteiz, Springer Berlin Heidelberg, 2012, pp. 216-223.

[81] O. Banos, J. M. Galvez, M. Damas, H. Pomares and I. Rojas, "Window Size Impact in Human Activity Recognition," *SENSORS,* vol. 14, no. 4, pp. 6474-6499, 2014.

[82] F. Miao, Y. He, J. Liu, Y. Li and I. Ayoola, "Identifying typical physical activity on smartphone with varying positions and orientations," *BioMedical Engineering OnLine,* pp. 14-32, 2015.

[83] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten and P. J. M. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *SENSORS,* vol. 14, no. 6, pp. 10146-10176, 2014.

ANNEX I: STRUCTURES

STRUCT 1

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **7369**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

TABLE 1.1: ACCURACY

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.900	0.003
Downstairs	0.930	0.004
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.993	0.000
Accuracy	99.2943%	

TABLE 1.2: CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2913	1	0	0	0	0
	Jog	0	646	0	0	0	0
	Up	0	1	269	29	0	0
	Down	0	0	21	278	0	0
	Sit	0	0	0	0	300	0
	Stand	0	0	0	0	0	2911

STRUCT 2

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **7369**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

TABLE 1.3: ACCURACY

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	0.998	0.000
Upstairs	0.896	0.005
Downstairs	0.890	0.005
Sitting	0.997	0.000
Standing	1.000	0.000
Average	0.991	0.000
Accuracy	99.0908 %	

TABLE 1.4: CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2913	1	0	0	0	0
	Jog	0	645	1	0	0	0
	Up	0	0	268	31	0	0
	Down	0	0	33	266	0	0
	Sit	0	0	0	1	299	0
	Stand	0	0	0	0	0	2911

STRUCT 3

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **7369**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

TABLE 1.5: ACCURACY

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.950	0.002
Downstairs	0.943	0.002
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.996	0.000
Accuracy	99.5657 %	

TABLE 1.6: CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2914	0	0	0	0	0
	Jog	0	646	0	0	0	0
	Up	0	0	284	14	0	1
	Down	1	0	16	282	0	0
	Sit	0	0	0	0	300	0
	Stand	0	0	0	0	0	2911

STRUCT 4

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **7369**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.970	0.002
Downstairs	0.960	0.001
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.997	0.000
Accuracy	99.715 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2914	0	0	0	0	0
	Jog	0	646	0	0	0	0
	Up	0	0	290	9	0	0
	Down	0	0	12	287	0	0
	Sit	0	0	0	0	300	0
	Stand	0	0	0	0	0	2911

STRUCT 5

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **7369**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	0.998	0.000
Upstairs	0.967	0.002
Downstairs	0.960	0.001
Sitting	1.000	0.000
Standing	0.999	0.000
Average	0.997	0.000
Accuracy	99.6879 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2913	1	0	0	0	0
	Jog	0	646	0	0	0	0
	Up	0	0	289	10	0	0
	Down	0	0	12	287	0	0
	Sit	0	0	0	0	300	0
	Stand	0	0	0	0	0	2911

STRUCT 6

VARIABLES

CODE: ***MOBIFALL***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **10343**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.999	0.001
Jogging	0.999	0.000
Upstairs	0.974	0.003
Downstairs	0.974	0.002
Sitting	0.991	0.000
Standing	1.000	0.000
Average	0.994	0.001
Accuracy	99.4102 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	4138	0	4	1	0	0
	Jog	0	3334	2	0	0	0
	Up	5	2	1021	20	0	0
	Down	1	1	20	809	0	0
	Sit	0	0	3	1	532	1
	Stand	0	0	0	0	0	448

STRUCT 7

VARIABLES

CODE: ***MOBIFALL***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **10343**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.993	0.005
Jogging	0.994	0.003
Upstairs	0.879	0.014
Downstairs	0.860	0.012
Sitting	0.983	0.001
Standing	0.989	0.001
Average	0.971	0.005
Accuracy	97.0511 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	4116	11	8	4	4	0
	Jog	14	3315	6	1	0	0
	Up	7	7	921	106	1	6
	Down	4	0	112	705	0	0
	Sit	4	0	1	0	528	4
	Stand	0	0	3	1	1	443

STRUCT 8

VARIABLES

CODE: ***MOBIFALL***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **10343**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.988	0.007
Jogging	0.993	0.004
Upstairs	0.848	0.018
Downstairs	0.818	0.015
Sitting	0.983	0.001
Standing	0.989	0.001
Average	0.962	0.007
Accuracy	96.181 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	4094	17	18	11	3	0
	Jog	16	3314	3	3	0	0
	Up	21	7	889	131	0	0
	Down	4	3	144	680	0	0
	Sit	0	0	3	1	528	5
	Stand	0	0	1	0	4	443

STRUCT 9

VARIABLES

CODE: ***MOBIFALL***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **10343**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.994	0.003
Jogging	0.995	0.000
Upstairs	0.971	0.006
Downstairs	0.949	0.004
Sitting	0.989	0.000
Standing	0.998	0.000
Average	0.988	0.002
Accuracy	98.8398 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	4118	1	14	9	0	1
	Jog	15	3320	1	0	0	0
	Up	2	0	1018	28	0	0
	Down	1	0	41	789	0	0
	Sit	3	0	0	0	531	3
	Stand	0	0	0	0	1	447

STRUCT 10

VARIABLES

CODE: ***MOBIFALL***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **10343**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.998	0.002
Jogging	0.997	0.001
Upstairs	0.998	0.012
Downstairs	0.874	0.014
Sitting	0.987	0.000
Standing	0.991	0.001
Average	0.974	0.003
Accuracy	97.3992 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	4133	3	5	2	0	0
	Jog	5	3327	3	1	0	0
	Up	5	1	914	126	0	2
	Down	1	0	104	726	0	0
	Sit	2	0	0	1	530	4
	Stand	0	0	1	1	2	444

STRUCT 11

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5268**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.985	0.030
Jogging	0.992	0.006
Upstairs	0.793	0.015
Downstairs	0.811	0.014
Sitting	0.870	0.002
Standing	0.913	0.006
Average	0.943	0.017
Accuracy	94.3242 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2055	1	15	16	0	0
	Jog	5	1670	5	4	0	0
	Up	50	20	438	44	0	0
	Down	38	1	44	355	0	0
	Sit	2	0	2	0	240	32
	Stand	1	0	7	2	10	211

STRUCT 12

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5268**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.926	0.059
Jogging	0.964	0.014
Upstairs	0.638	0.040
Downstairs	0.616	0.033
Sitting	0.971	0.001
Standing	0.983	0.001
Average	0.887	0.035
Accuracy	88.7054 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1932	13	74	65	1	2
	Jog	22	1624	28	10	0	0
	Up	88	28	352	83	1	0
	Down	78	10	80	270	0	0
	Sit	0	0	4	0	268	4
	Stand	0	0	1	1	2	227

STRUCT 13

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5268**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.940	0.079
Jogging	0.979	0.011
Upstairs	0.578	0.035
Downstairs	0.546	0.029
Sitting	0.917	0.003
Standing	0.926	0.004
Average	0.880	0.041
Accuracy	87.9841 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1961	6	63	54	0	3
	Jog	21	1649	5	9	0	0
	Up	121	31	319	76	1	4
	Down	105	3	90	239	1	0
	Sit	4	0	3	2	253	14
	Stand	0	0	4	1	12	214

STRUCT 14

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5268**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **Multilayer perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.952	0.045
Jogging	0.982	0.012
Upstairs	0.725	0.024
Downstairs	0.708	0.024
Sitting	0.924	0.004
Standing	0.913	0.004
Average	0.914	0.026
Accuracy	91.4389 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1987	6	40	53	1	0
	Jog	14	1654	13	3	0	0
	Up	63	30	400	57	1	1
	Down	61	6	61	310	0	0
	Sit	4	0	0	0	255	17
	Stand	1	0	0	1	18	211

STRUCT 15

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5268**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.950	0.052
Jogging	0.980	0.009
Upstairs	0.707	0.033
Downstairs	0.619	0.025
Sitting	0.924	0.002
Standing	0.935	0.003
Average	0.905	0.029
Accuracy	90.4708 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1983	6	51	46	0	1
	Jog	15	1651	13	5	0	0
	Up	73	20	390	69	0	0
	Down	76	7	84	271	0	0
	Sit	1	0	5	0	255	15
	Stand	0	0	2	2	11	216

STRUCT 16

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4169**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.990	0.032
Jogging	1.000	0.001
Upstairs	0.746	0.015
Downstairs	0.686	0.013
Sitting	0.943	0.001
Standing	0.995	0.002
Average	0.950	0.014
Accuracy	94.9868 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1450	1	5	8	0	0
	Jog	0	346	0	0	0	0
	Up	38	2	223	36	0	0
	Down	46	0	48	205	0	0
	Sit	2	0	2	7	283	6
	Stand	0	0	4	0	4	1453

STRUCT 17

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4169**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.972	0.032
Jogging	0.974	0.002
Upstairs	0.686	0.019
Downstairs	0.692	0.017
Sitting	0.987	0.002
Standing	0.999	0.000
Average	0.942	0.014
Accuracy	94.1953 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1423	6	18	13	4	0
	Jog	6	337	2	1	0	0
	Up	44	1	205	49	0	0
	Down	36	1	55	207	0	0
	Sit	1	0	0	2	296	1
	Stand	0	0	0	0	2	1459

STRUCT 18

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4169**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.977	0.019
Jogging	0.988	0.002
Upstairs	0.773	0.015
Downstairs	0.833	0.010
Sitting	0.983	0.002
Standing	0.996	0.000
Average	0.960	0.009
Accuracy	95.9942 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1430	3	23	6	1	1
	Jog	4	342	0	0	0	0
	Up	33	4	231	31	0	0
	Down	12	1	34	249	3	0
	Sit	2	1	2	0	295	0
	Stand	1	0	0	1	4	1455

STRUCT 19

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4169**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.983	0.020
Jogging	0.986	0.002
Upstairs	0.796	0.011
Downstairs	0.843	0.011
Sitting	0.947	0.002
Standing	0.998	0.002
Average	0.962	0.009
Accuracy	96.2341 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1439	3	10	12	0	0
	Jog	5	341	0	0	0	0
	Up	25	3	238	27	3	3
	Down	20	1	25	252	1	0
	Sit	3	1	5	5	284	2
	Stand	0	0	1	0	2	1458

STRUCT 20

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4169**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.978	0.017
Jogging	0.991	0.001
Upstairs	0.829	0.013
Downstairs	0.843	0.010
Sitting	0.973	0.001
Standing	0.998	0.000
Average	0.965	0.008
Accuracy	96.5459 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1432	3	18	10	1	0
	Jog	3	343	0	0	0	0
	Up	26	1	248	24	0	0
	Down	15	1	30	252	1	0
	Sit	1	0	3	3	292	1
	Stand	0	0	0	2	1	1458

STRUCT 21

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4313**
USERS: **29**
CLASIFICATION ALGORITHM: **IBK**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 2,4,9,19,20,25,32**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.986	0.032
Jogging	0.990	0.007
Upstairs	0.789	0.017
Downstairs	0.824	0.014
Sitting	0.847	0.001
Standing	0.904	0.006
Average	0.941	0.018
Accuracy	94.134 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1666	1	10	12	0	0
	Jog	6	1372	5	3	0	0
	Up	46	18	385	39	0	0
	Down	29	2	37	318	0	0
	Sit	2	0	2	0	150	23
	Stand	0	0	11	1	6	169

STRUCT 22

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4313**
USERS: **29**
CLASIFICATION ALGORITHM: **J48**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 2,4,9,19,20,25,32**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.929	0.060
Jogging	0.974	0.012
Upstairs	0.645	0.041
Downstairs	0.588	0.036
Sitting	0.994	0.000
Standing	0.984	0.001
Average	0.886	0.035
Accuracy	88.5926 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1569	4	59	56	0	1
	Jog	4	1350	21	11	0	0
	Up	74	24	315	73	1	1
	Down	78	7	74	227	0	0
	Sit	0	0	0	0	176	1
	Stand	1	0	1	0	1	184

STRUCT 23

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4313**
USERS: **29**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 2,4,9,19,20,25,32**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.943	0.077
Jogging	0.982	0.011
Upstairs	0.584	0.038
Downstairs	0.565	0.029
Sitting	0.932	0.004
Standing	0.866	0.004
Average	0.877	0.041
Accuracy	87.7348 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1593	4	50	41	0	1
	Jog	11	1361	9	4	0	1
	Up	104	26	285	70	0	3
	Down	83	2	80	218	1	2
	Sit	2	0	0	0	165	10
	Stand	3	0	6	0	16	162

STRUCT 24

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4313**
USERS: **29**
CLASIFICATION ALGORITHM: **Multilayer perceptron**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 2,4,9,19,20,25,32**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.967	0.037
Jogging	0.985	0.011
Upstairs	0.748	0.021
Downstairs	0.733	0.022
Sitting	0.915	0.002
Standing	0.957	0.005
Average	0.924	0.023
Accuracy	92.4415 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1633	2	23	31	0	0
	Jog	11	1365	7	2	1	0
	Up	42	24	365	52	1	4
	Down	44	5	51	283	0	3
	Sit	0	0	0	0	162	15
	Stand	1	0	1	1	5	179

STRUCT 25

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4313**
USERS: **29**
CLASIFICATION ALGORITHM: **LMT**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 2,4,9,19,20,25,32**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.950	0.051
Jogging	0.982	0.010
Upstairs	0.742	0.026
Downstairs	0.661	0.026
Sitting	0.949	0.002
Standing	0.941	0.003
Average	0.911	0.029
Accuracy	91.0503 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1605	4	40	39	0	1
	Jog	6	1361	9	10	0	0
	Up	54	20	362	51	0	1
	Down	75	5	51	255	0	0
	Sit	0	0	0	0	168	9
	Stand	0	0	1	2	8	176

STRUCT 26

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5841**
USERS: **40 (range 2 - 48)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.879	0.000
Downstairs	0.942	0.005
Sitting	1.000	0.002
Standing	1.000	0.000
Average	0.992	0.000
Accuracy	99.2467 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2332	1	0	0	0	0
	Jog	0	518	0	0	0	0
	Up	0	1	210	28	0	0
	Down	0	0	14	226	0	0
	Sit	0	0	0	3	240	0
	Stand	0	0	0	0	0	2271

STRUCT 27

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5841**
USERS: **40 (range 2 - 48)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.001
Jogging	0.998	0.000
Upstairs	0.912	0.005
Downstairs	0.879	0.004
Sitting	0.996	0.000
Standing	0.999	0.000
Average	0.991	0.001
Accuracy	99.0584 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2332	1	0	0	0	0
	Jog	0	517	1	0	0	0
	Up	0	0	218	21	0	0
	Down	0	0	29	211	0	0
	Sit	0	0	0	1	239	0
	Stand	2	0	0	0	0	2269

STRUCT 28

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5841**
USERS: **40 (range 2 - 48)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.975	0.002
Downstairs	0.942	0.001
Sitting	0.996	0.000
Standing	0.999	0.000
Average	0.996	0.000
Accuracy	99.6062 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2333	1	0	0	0	0
	Jog	0	518	1	0	0	0
	Up	0	1	233	5	0	0
	Down	0	0	14	226	0	0
	Sit	0	0	0	0	239	1
	Stand	0	0	0	0	2	2269

STRUCT 29

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5841**
USERS: **40 (range 2 - 48)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.958	0.002
Downstairs	0.963	0.002
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.997	0.000
Accuracy	99.6747 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2333	0	0	0	0	0
	Jog	0	518	0	0	0	0
	Up	0	0	229	10	0	0
	Down	0	0	9	231	0	0
	Sit	0	0	0	0	240	0
	Stand	0	0	0	0	0	2271

STRUCT 30

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **5841**
USERS: **40 (range 2 - 48)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.001
Jogging	0.998	0.000
Upstairs	0.954	0.002
Downstairs	0.950	0.002
Sitting	1.000	0.000
Standing	0.999	0.000
Average	0.995	0.000
Accuracy	99.5378 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2332	1	0	0	0	0
	Jog	0	517	0	1	0	0
	Up	0	0	228	11	0	0
	Down	0	0	12	228	0	0
	Sit	0	0	0	0	240	0
	Stand	2	0	0	0	0	2269

STRUCT 31

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **42153**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.001
Jogging	0.991	0.000
Upstairs	0.899	0.008
Downstairs	0.887	0.007
Sitting	0.995	0.000
Standing	1.000	0.000
Average	0.985	0.002
Accuracy	98.4888 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14908	1	1	0	0	4
	Jog	37	4206	0	3	0	0
	Up	1	2	2387	298	3	0
	Down	0	1	261	2420	8	1
	Sit	0	0	0	13	2687	0
	Stand	0	0	0	1	2	14908

STRUCT 32

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **42153**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	0.999	0.000
Upstairs	0.848	0.011
Downstairs	0.845	0.010
Sitting	0.994	0.000
Standing	1.000	0.000
Average	0.980	0.001
Accuracy	97.9812 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14913	1	0	0	0	0
	Jog	0	4242	3	1	0	0
	Up	0	0	2281	395	15	0
	Down	0	0	416	2273	2	0
	Sit	0	0	15	2	2683	0
	Stand	1	0	0	0	0	14910

STRUCT 33

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **42153**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	0.999	0.000
Upstairs	0.856	0.008
Downstairs	0.880	0.010
Sitting	0.996	0.000
Standing	1.000	0.000
Average	0.983	0.001
Accuracy	98.273 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2914	0	0	0	0	0
	Jog	0	646	0	0	0	0
	Up	0	0	284	14	0	1
	Down	1	0	16	282	0	0
	Sit	0	0	0	0	300	0
	Stand	0	0	0	0	0	2911

STRUCT 34

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **42153**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.886	0.006
Downstairs	0.909	0.008
Sitting	0.998	0.000
Standing	1.000	0.000
Average	0.987	0.001
Accuracy	98.6715 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14913	0	0	0	0	1
	Jog	0	4246	0	0	0	0
	Up	0	0	2384	299	8	0
	Down	0	0	244	2445	2	0
	Sit	0	0	5	1	2694	0
	Stand	0	0	0	0	0	14911

STRUCT 35

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **42153**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	0.999	0.000
Upstairs	0.876	0.002
Downstairs	0.885	0.008
Sitting	0.997	0.008
Standing	1.000	0.000
Average	0.984	0.001
Accuracy	98.4461 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14913	1	0	0	0	0
	Jog	0	4242	3	1	0	0
	Up	0	0	2358	331	2	0
	Down	0	0	309	2381	1	0
	Sit	0	0	7	0	2693	0
	Stand	0	0	0	0	0	14911

STRUCT 36

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **40**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **27180**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.957	0.094
Jogging	0.978	0.012
Upstairs	0.541	0.029
Downstairs	0.547	0.028
Sitting	0.902	0.002
Standing	0.921	0.005
Average	0.876	0.047
Accuracy	87.638 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	10128	23	213	213	1	1
	Jog	81	8340	67	37	0	0
	Up	812	149	1627	411	3	6
	Down	669	56	377	1339	3	3
	Sit	3	0	12	7	1284	118
	Stand	2	0	30	15	48	1102

STRUCT 37

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **40**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **27180**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.912	0.064
Jogging	0.970	0.017
Upstairs	0.594	0.043
Downstairs	0.599	0.039
Sitting	0.981	0.001
Standing	0.982	0.001
Average	0.874	0.035
Accuracy	87.362 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	9649	58	448	420	2	2
	Jog	78	8271	121	55	0	0
	Up	539	185	1787	481	8	8
	Down	453	67	456	1465	2	4
	Sit	0	0	7	8	1397	12
	Stand	0	0	7	2	12	1176

STRUCT 38

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **40**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **27180**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.938	0.150
Jogging	0.970	0.021
Upstairs	0.349	0.034
Downstairs	0.405	0.026
Sitting	0.949	0.001
Standing	0.971	0.003
Average	0.837	0.071
Accuracy	83.6718 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	9918	54	355	248	1	3
	Jog	193	8269	33	30	0	0
	Up	1335	260	1051	349	5	8
	Down	962	74	417	990	1	3
	Sit	5	0	3	4	1352	60
	Stand	1	0	4	1	29	1162

STRUCT 39

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **40**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **27180**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **Multilayer perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.933	0.072
Jogging	0.978	0.016
Upstairs	0.608	0.035
Downstairs	0.602	0.029
Sitting	0.952	0.002
Standing	0.971	0.003
Average	0.884	0.040
Accuracy	88.3922 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	9872	26	385	288	4	4
	Jog	80	8335	63	46	1	0
	Up	580	217	1829	369	2	11
	Down	529	54	387	1472	4	1
	Sit	3	0	5	4	1355	57
	Stand	2	0	3	1	29	1162

STRUCT 40

VARIABLES

CODE: ***WISDM***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **40**
STEP: **40**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **27180**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.950	0.082
Jogging	0.970	0.014
Upstairs	0.602	0.029
Downstairs	0.617	0.024
Sitting	0.974	0.001
Standing	0.981	0.001
Average	0.890	0.042
Accuracy	89.0397 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	10054	35	285	202	1	2
	Jog	117	8266	99	43	0	0
	Up	668	176	1811	347	1	5
	Down	572	50	309	1509	3	4
	Sit	0	0	6	8	1387	23
	Stand	1	0	7	2	13	1174

STRUCT 41

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **38709**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.993	0.001
Downstairs	0.982	0.000
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.999	0.000
Accuracy	99.8837 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14764	0	0	0	0	0
	Jog	0	3796	0	0	0	0
	Up	0	0	1781	13	0	0
	Down	0	0	32	1762	0	0
	Sit	0	0	0	0	1800	0
	Stand	0	0	0	0	0	14761

STRUCT 42

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **38709**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.930	0.004
Downstairs	0.921	0.003
Sitting	0.999	0.000
Standing	1.000	0.000
Average	0.993	0.000
Accuracy	99.2973 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14763	1	0	0	0	0
	Jog	0	3795	0	1	0	0
	Up	0	0	1668	126	0	0
	Down	0	0	142	1652	0	0
	Sit	0	0	0	2	1798	0
	Stand	0	0	0	0	0	14761

STRUCT 43

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **38709**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LOGISTIC**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.967	0.002
Downstairs	0.966	0.002
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.997	0.000
Accuracy	99.6874 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14764	0	0	0	0	0
	Jog	0	3796	0	0	0	0
	Up	0	0	1734	60	0	0
	Down	0	0	61	1733	0	0
	Sit	0	0	0	0	1800	0
	Stand	0	0	0	0	0	14761

STRUCT 44

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **38709**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **MULTILAYER PERCEPTRON**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.991	0.001
Downstairs	0.981	0.000
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.999	0.000
Accuracy	99.8682 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14764	0	0	0	0	0
	Jog	0	3796	0	0	0	0
	Up	0	0	1777	16	0	1
	Down	0	0	34	1760	0	0
	Sit	0	0	0	0	1800	0
	Stand	0	0	0	0	0	14761

STRUCT 45

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **38709**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.974	0.001
Downstairs	0.976	0.001
Sitting	0.999	0.000
Standing	1.000	0.000
Average	0.998	0.000
Accuracy	99.7597 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	14763	1	0	0	0	0
	Jog	0	3795	0	1	0	0
	Up	0	0	1747	47	0	0
	Down	0	0	43	1751	0	0
	Sit	0	0	1	0	1799	0
	Stand	0	0	0	0	0	14761

STRUCT 46

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **52938**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	0.999	0.000
Upstairs	0.992	0.001
Downstairs	0.991	0.001
Sitting	0.999	0.000
Standing	0.999	0.000
Average	0.998	0.000
Accuracy	99.7922 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	20922	0	5	4	0	0
	Jog	3	16843	2	4	0	0
	Up	8	1	5548	34	2	1
	Down	5	0	35	4445	0	0
	Sit	0	0	2	0	2762	2
	Stand	0	0	2	0	0	2308

STRUCT 47

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **52938**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.998	0.002
Jogging	0.998	0.001
Upstairs	0.939	0.006
Downstairs	0.937	0.007
Sitting	0.996	0.000
Standing	0.996	0.000
Average	0.986	0.002
Accuracy	98.6286 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	20889	21	12	6	3	0
	Jog	30	16812	3	6	1	0
	Up	23	8	5253	306	1	3
	Down	2	2	278	4202	0	1
	Sit	2	3	0	1	2756	4
	Stand	0	0	2	0	8	2300

STRUCT 48

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **52938**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **LOGISTIC**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.990	0.010
Jogging	0.992	0.002
Upstairs	0.808	0.022
Downstairs	0.796	0.019
Sitting	0.996	0.000
Standing	0.997	0.000
Average	0.955	0.008
Accuracy	95.5776 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	20722	40	152	14	2	1
	Jog	115	16711	11	13	2	0
	Up	159	24	4518	890	0	3
	Down	42	2	868	3570	3	0
	Sit	3	0	1	3	2756	3
	Stand	0	0	2	0	4	2304

STRUCT 49

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **52938**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **MULTILAYER PERCEPTRON**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.995	0.001
Jogging	0.999	0.000
Upstairs	0.974	0.003
Downstairs	0.978	0.004
Sitting	0.996	0.000
Standing	0.997	0.000
Average	0.993	0.001
Accuracy	99.2614 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	20822	7	44	42	1	15
	Jog	17	16835	0	0	0	0
	Up	3	3	5447	138	2	1
	Down	0	0	100	4385	0	0
	Sit	3	2	0	0	2755	6
	Stand	0	0	1	1	5	2303

STRUCT 50

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **52938**
USERS: **36 (range 1 - 36)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.998	0.001
Jogging	0.990	0.000
Upstairs	0.892	0.008
Downstairs	0.912	0.012
Sitting	0.997	0.001
Standing	0.998	0.000
Average	0.980	0.002
Accuracy	97.9599 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	20883	4	7	2	35	0
	Jog	20	16828	1	3	0	0
	Up	3	0	4991	598	0	2
	Down	7	0	386	4092	0	0
	Sit	1	0	3	0	2758	4
	Stand	0	0	4	0	0	2306

STRUCT 51

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **40Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **82171**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	0.998	0.000
Upstairs	0.970	0.003
Downstairs	0.948	0.002
Sitting	0.999	0.000
Standing	1.000	0.000
Average	0.995	0.000
Accuracy	99.4986 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	29760	0	0	0	0	1
	Jog	14	8269	0	0	0	0
	Up	0	0	4642	142	0	0
	Down	0	0	249	4533	2	0
	Sit	0	0	4	0	4796	0
	Stand	0	0	0	0	0	29759

STRUCT 52

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **40Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **82171**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.918	0.005
Downstairs	0.917	0.005
Sitting	0.999	0.000
Standing	1.000	0.000
Average	0.990	0.001
Accuracy	99.0228 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	29760	1	0	0	0	0
	Jog	0	8280	0	3	0	0
	Up	0	0	4392	390	2	0
	Down	0	0	398	4385	1	0
	Sit	0	0	6	0	4794	0
	Stand	2	0	0	0	0	29757

STRUCT 53

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **40Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **82171**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LOGISTIC**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.901	0.005
Downstairs	0.914	0.006
Sitting	0.999	0.000
Standing	1.000	0.000
Average	0.989	0.001
Accuracy	98.9145 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	29759	0	0	0	0	2
	Jog	1	8282	0	0	0	0
	Up	0	0	4312	470	2	0
	Down	0	0	409	4371	4	0
	Sit	0	0	2	0	4796	2
	Stand	0	0	0	0	0	29759

STRUCT 54

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **40Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **82171**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **MULTILAYER PERCEPTRON**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.953	0.003
Downstairs	0.947	0.003
Sitting	0.999	0.000
Standing	1.000	0.000
Average	0.994	0.000
Accuracy	99.4171 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	29761	0	0	0	0	0
	Jog	0	8283	0	0	0	0
	Up	0	0	4560	223	1	0
	Down	0	0	251	4532	1	0
	Sit	0	0	2	1	4797	0
	Stand	0	0	0	0	0	29759

STRUCT 55

VARIABLES

CODE: ***PFS***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **40Hz**
WINDOW SIZE: **100**
STEP: **20**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **82171**
USERS: **50 (range 2 - 57)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.912	0.004
Downstairs	0.927	0.005
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.991	0.001
Accuracy	99.0593 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	29760	1	0	0	0	0
	Jog	0	8280	0	3	0	0
	Up	0	0	4362	422	0	0
	Down	0	0	347	4437	0	0
	Sit	0	0	0	0	4800	0
	Stand	0	0	0	0	0	29759

STRUCT 56

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4184**
USERS: **29 (range 1 - 29)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 30,31,32,33,34,35,36**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.996	0.004
Jogging	0.999	0.001
Upstairs	0.932	0.010
Downstairs	0.906	0.007
Sitting	0.975	0.000
Standing	0.994	0.000
Average	0.982	0.004
Accuracy	98.1836 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1672	0	4	2	0	0
	Jog	1	1355	1	0	0	0
	Up	6	2	410	22	0	0
	Down	3	0	29	308	0	0
	Sit	1	0	2	1	197	1
	Stand	0	0	0	1	0	166

STRUCT 57

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4184**
USERS: **29 (range 1 - 29)**
CLASIFICATION ALGORITHM: **J48**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 30,31,32,33,34,35,36**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.994	0.006
Jogging	0.991	0.003
Upstairs	0.655	0.013
Downstairs	0.874	0.016
Sitting	0.970	0.001
Standing	0.994	0.000
Average	0.967	0.006
Accuracy	96.7256 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1668	7	0	2	1	0
	Jog	9	1345	3	0	0	0
	Up	4	1	375	59	0	1
	Down	0	0	43	297	0	0
	Sit	3	0	2	0	196	1
	Stand	0	0	0	0	1	166

STRUCT 58

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4184**
USERS: **29 (range 1 - 29)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 30,31,32,33,34,35,36**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.983	0.012
Jogging	0.994	0.002
Upstairs	0.795	0.023
Downstairs	0.774	0.021
Sitting	0.975	0.002
Standing	0.970	0.001
Average	0.949	0.010
Accuracy	94.9092 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1650	4	14	9	1	0
	Jog	4	1349	4	0	0	0
	Up	16	1	350	70	1	2
	Down	8	1	67	263	0	1
	Sit	1	0	1	1	197	2
	Stand	0	0	0	0	5	162

STRUCT 59

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4184**
USERS: **29 (range 1 - 29)**
CLASIFICATION ALGORITHM: **Multilayer perceptron**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 30,31,32,33,34,35,36**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.998	0.004
Jogging	0.996	0.000
Upstairs	0.925	0.009
Downstairs	0.915	0.019
Sitting	0.980	0.000
Standing	0.994	0.000
Average	0.982	0.003
Accuracy	98.1836 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1674	0	3	1	0	0
	Jog	5	1352	0	0	0	0
	Up	2	0	407	31	0	0
	Down	0	0	29	311	0	0
	Sit	2	0	0	0	198	2
	Stand	0	0	0	1	0	166

STRUCT 60

VARIABLES

CODE: ***PFS***

DATA: ***WISDM***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4184**
USERS: **29 (range 1 - 29)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS: **(users) RemoveWithValues -S 0.0 -C 1 -L 30,31,32,33,34,35,36**
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.992	0.004
Jogging	0.996	0.002
Upstairs	0.857	0.015
Downstairs	0.850	0.016
Sitting	0.990	0.001
Standing	0.994	0.000
Average	0.967	0.005
Accuracy	96.7256 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1664	3	8	2	1	0
	Jog	5	1351	1	0	0	0
	Up	1	3	377	58	1	0
	Down	3	0	48	289	0	0
	Sit	1	0	0	1	200	0
	Stand	0	0	0	0	1	166

STRUCT 61

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **2425**
USERS: **29 (range 2 - 36)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.985	0.036
Jogging	1.000	0.001
Upstairs	0.728	0.016
Downstairs	0.661	0.015
Sitting	0.960	0.000
Standing	0.993	0.002
Average	0.946	0.015
Accuracy	94.5567 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	838	1	5	7	0	0
	Jog	0	203	0	0	0	0
	Up	21	2	126	24	0	0
	Down	31	0	28	115	0	0
	Sit	3	0	0	1	167	3
	Stand	1	0	3	1	1	844

STRUCT 62

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **2425**
USERS: **29 (range 2 - 36)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.965	0.026
Jogging	0.970	0.002
Upstairs	0.763	0.018
Downstairs	0.730	0.017
Sitting	0.989	0.001
Standing	0.998	0.001
Average	0.947	0.012
Accuracy	94.7216 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	821	1	18	11	0	0
	Jog	3	197	1	2	0	0
	Up	16	1	132	24	0	0
	Down	22	3	22	127	0	0
	Sit	0	0	0	1	172	1
	Stand	0	0	0	0	2	848

STRUCT 63

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **2425**
USERS: **29 (range 2 - 36)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.971	0.020
Jogging	0.990	0.002
Upstairs	0.763	0.016
Downstairs	0.805	0.016
Sitting	0.971	0.000
Standing	0.995	0.001
Average	0.954	0.010
Accuracy	95.4227 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	826	3	11	11	0	0
	Jog	2	201	0	0	0	0
	Up	17	1	132	23	0	0
	Down	12	0	22	140	0	0
	Sit	0	1	3	0	169	1
	Stand	1	0	0	2	1	846

STRUCT 64

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **2425**
USERS: **29 (range 2 - 36)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.982	0.026
Jogging	1.000	0.002
Upstairs	0.803	0.012
Downstairs	0.776	0.009
Sitting	0.948	0.002
Standing	0.998	0.001
Average	0.959	0.011
Accuracy	95.9175 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	836	1	8	6	0	0
	Jog	0	203	0	0	0	0
	Up	19	3	139	11	0	1
	Down	17	1	18	135	3	0
	Sit	4	0	2	3	165	0
	Stand	1	0	0	0	1	848

STRUCT 65

VARIABLES

CODE: ***WISDM***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **200**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **2425**
USERS: **29 (range 2 - 36)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.980	0.022
Jogging	1.000	0.001
Upstairs	0.780	0.013
Downstairs	0.782	0.012
Sitting	0.983	0.001
Standing	0.998	0.001
Average	0.960	0.010
Accuracy	95.9588 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	834	0	10	7	0	0
	Jog	0	203	0	0	0	0
	Up	15	2	135	21	0	0
	Down	18	0	20	136	0	0
	Sit	1	1	0	0	171	1
	Stand	0	0	0	0	2	848

STRUCT 66

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4281**
USERS: 29 **(range 2 - 36)**
CLASIFICATION ALGORITHM: **IBK**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	TPRate	FPRate
Walking	0.999	0.000
Jogging	1.000	0.001
Upstairs	0.855	0.003
Downstairs	0.925	0.006
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.991	0.000
Accuracy	99.089 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1691	1	0	0	0	0
	Jog	0	377	0	0	0	0
	Up	0	1	148	24	0	0
	Down	0	0	13	161	0	0
	Sit	0	0	0	0	174	0
	Stand	0	0	0	0	0	1691

STRUCT 67

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4281**
USERS: 29 **(range 2 - 36)**
CLASIFICATION ALGORITHM: **J48**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.999	0.000
Jogging	0.997	0.000
Upstairs	0.884	0.006
Downstairs	0.862	0.005
Sitting	0.994	0.000
Standing	0.999	0.000
Average	0.989	0.001
Accuracy	98.8788 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1691	1	0	0	0	0
	Jog	0	376	0	1	0	0
	Up	0	0	153	20	0	0
	Down	0	0	24	150	0	0
	Sit	0	0	0	1	173	0
	Stand	1	0	0	0	0	1690

STRUCT 68

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4281**
USERS: 29 **(range 2 - 36)**
CLASIFICATION ALGORITHM: **Logistic**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	0.999	0.000
Jogging	0.995	0.001
Upstairs	0.971	0.002
Downstairs	0.937	0.001
Sitting	0.994	0.001
Standing	1.000	0.001
Average	0.995	0.001
Accuracy	99.5095 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1691	0	0	0	1	0
	Jog	0	375	1	1	0	0
	Up	0	1	168	3	1	0
	Down	1	1	7	163	1	1
	Sit	0	0	0	0	173	1
	Stand	0	0	0	1	0	1690

STRUCT 69

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4281**
USERS: 29 **(range 2 - 36)**
CLASIFICATION ALGORITHM: **Multilayer Perceptron**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.936	0.002
Downstairs	0.943	0.003
Sitting	1.000	0.000
Standing	1.000	0.000
Average	0.995	0.000
Accuracy	99.5095 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	2914	0	0	0	0	0
	Jog	0	646	0	0	0	0
	Up	0	0	290	9	0	0
	Down	0	0	12	287	0	0
	Sit	0	0	0	0	300	0
	Stand	0	0	0	0	0	2911

STRUCT 70

VARIABLES

CODE: ***MOBIFALL***

DATA: ***MOBIFALL***

CODE PARAMETERS

SAMPLING RATE: **20Hz**
WINDOW SIZE: **200**
STEP: **100**

WEKA PARAMETERS AND STATISTICS

INSTANCES: **4281**
USERS: 29 **(range 2 - 36)**
CLASIFICATION ALGORITHM: **LMT**
FILTERS:
FOLDS: **10**

RUSULTS

ACCURACY TABLE

CLASS	**TPRate**	**FPRate**
Walking	1.000	0.000
Jogging	1.000	0.000
Upstairs	0.936	0.002
Downstairs	0.948	0.003
Sitting	0.994	0.000
Standing	1.000	0.000
Average	0.995	0.000
Accuracy	99.5095 %	

CONFUSION MATRIX

		PREDICTED CLASS					
		Walk	**Jog**	**Up**	**Down**	**Sit**	**Stand**
ACTUAL CLASS	**Walk**	1692	0	0	0	0	0
	Jog	0	377	0	0	0	0
	Up	0	0	162	11	0	0
	Down	0	0	9	165	0	0
	Sit	0	0	0	1	173	0
	Stand	0	0	0	0	0	1691