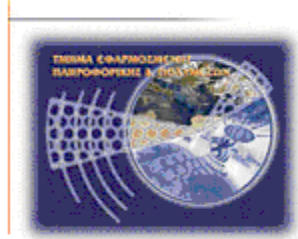




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή εργασία

**Εφαρμογή ηλεκτρονικής υγείας για την
παρακολούθηση ασθενών με τη χρήση έξυπνων
κινητών**

Βάβουλας Γεώργιος ΑΜ: 1859

Ηράκλειο – 28/3/2014

Επόπτης Καθηγητής: Τσικνάκης Μανώλης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή Μανώλη Τσικνάκη για την βοήθεια και υποστήριξη που μου παρείχε καθ' όλη τη διάρκεια υλοποίησης της πτυχιακής μου εργασίας και πορείας μου ως προπτυχιακός φοιτητής του τμήματος. Επιπλέον ένα μεγάλο ευχαριστώ στους Ματθαίο Πεδιάδιτη και Μανώλη Σπανάκη για την βοήθεια και την καθοδήγησή τους. Φυσικά δεν θα μπορούσα να παραλείψω να ευχαριστήσω τους γονείς μου που με στηρίζουν σε κάθε μου βήμα. Ακόμα θα ήθελα να ευχαριστήσω τους Μανώλη Δασκαλάκη και Ρούλα Χατζάκη καθώς και τα μέλη του εργαστηρίου *Βιοϊατρικής Πληροφορικής και Υπηρεσιών Ηλεκτρονικής Υγείας*, Νεφέλη, Νίκη και Ελένη. Τέλος ένα μεγάλο ευχαριστώ στους εθελοντές που βοήθησαν στην καταγραφή δεδομένων.

Abstract

Currently, recognizing human activities is an important research topic; it is exploited widely and applied to many fields in real-life, especially health care or context aware applications. Research achievements are mainly focused on accurate recognition and classification of Activities of Daily Living (ADL) which are useful for developing smart eHealth and mHealth systems and services.

Falls is one of the biggest risks to the health and wellbeing of the elderly. Falls in the elderly are a major problem for today's society. Approximately one in every three adults aged 65 years old or older falls each year creating physical injuries and psychological harm. In addition significant economic effects result from this problem. According to World Health Organization in the year 2030 the estimation of injuries related to falls will increase by 100%. The ability therefore to distinguish between Activities of Daily Living (ADL) and falls is therefore an important problem.

The objective of the present thesis is twofold. Firstly to create a comprehensive data set of ADLs and falls performed by young volunteers under supervised conditions together with an application for smart phones using the Android operation system for the detailed annotation of the data set. The dataset contains signals recorded from the accelerometer and gyroscope sensors of a latest technology smartphone for four different types of falls and nine different activities of daily living. During this thesis an application using the android operating system for smartphones was developed. Using this application recordings were made which contain data from smartphone sensors. The application also supports the detailed annotation of the resulting dataset. Our second objective was to exploit this dataset for the comparative evaluation of the performance of three known fall detection algorithms that have previously been used in smartphone-based fall-detection. The results of this elaborate evaluation of fall detection are presented, utilizing the aforementioned dataset, and discussed in detail.

Σύνοψη

Σήμερα η αναγνώριση των ανθρώπινων δραστηριοτήτων της καθημερινότητας είναι ένα σημαντικό ερευνητικό θέμα. Αξιοποιείται ευρέως και εφαρμόζεται σε πολλούς τομείς στην πραγματική ζωή. Η έρευνα εστιάζει κυρίως στην ακριβή αναγνώριση και ταξινόμηση των Δραστηριοτήτων της Καθημερινής Ζωής (Activities of Daily Living - ADLs), που είναι χρήσιμα για την ανάπτυξη έξυπνων εφαρμογών και υπηρεσιών ηλεκτρονικής υγείας

Παράλληλα, οι πτώσεις στους ηλικιωμένους είναι ένα σημαντικό πρόβλημα για την κοινωνία. Περίπου ένας στους τρεις ηλικιωμένους από 65 ετών και άνω πέφτουν κάθε χρόνο με αποτέλεσμα τις σωματικές και ψυχολογικές βλάβες. Επίσης οι οικονομικές συνέπειες των πτώσεων είναι σημαντικές. Σύμφωνα με τον Παγκόσμιο οργανισμό υγείας το έτος 2030 η εκτίμηση των τραυματισμών που σχετίζονται με πτώσεις θα αυξηθεί κατά 100%. Η ικανότητα να ξεχωρίσουμε τις καθημερινές δραστηριότητες από τις πτώσεις είναι ένα σημαντικό πρόβλημα.

Οι στόχοι αυτής της πτυχιακής είναι δύο. Ο πρώτος είναι να παράξει ένα ολοκληρωμένο σύνολο δεδομένων που αποτελείται από προσομοιωμένες καθημερινές δραστηριότητες και πτώσεις χρησιμοποιώντας νεαρούς εθελοντές με την χρήση μίας εφαρμογής για έξυπνα κινητά τηλέφωνα και τον λεπτομερή σχολιασμό του συνόλου αυτού των δεδομένων. Το σύνολο δεδομένων περιλαμβάνει σήματα που καταγράφονται από τους αισθητήρες του επιταχυνσιόμετρου και γυροσκοπίου ενός έξυπνου κινητού τελευταίας τεχνολογίας για τέσσερις διαφορετικούς τύπους πτώσεων και εννέα διαφορετικές δραστηριότητες της καθημερινής ζωής. Κατά τη διάρκεια της πτυχιακής εργασίας αναπτύχθηκε μία εφαρμογή σε λειτουργικό σύστημα Android. Με την χρήση αυτής της εφαρμογής καταγράφηκαν σε αρχεία, δεδομένα από τους αισθητήρες ενός έξυπνου κινητού τηλεφώνου. Οι αισθητήρες που χρησιμοποιήθηκαν είναι το επιταχυνσιόμετρο, το γυροσκόπιο και ο αισθητήρας προσανατολισμού. Ο δεύτερος στόχος μας ήταν να εκμεταλλευτούμε αυτό το σύνολο δεδομένων για τη συγκριτική αξιολόγηση των επιδόσεων τριών γνωστών αλγορίθμων ανίχνευσης πτώσεων που έχουν προηγουμένως χρησιμοποιηθεί σε εφαρμογές έξυπνων κινητών. Σε δεύτερη φάση χρησιμοποιήθηκαν τεχνικές μηχανικής μάθησης για την ανίχνευση πτώσης αλλά και της κατηγοριοποίησης της πτώσης. Τα αποτελέσματα αυτής της αξιολόγησης, χρησιμοποιώντας το προαναφερθέν σύνολο δεδομένων, παρουσιάζονται και συζητούνται λεπτομερώς.

Πίνακας Περιεχομένων

Ευχαριστίες	ii
Abstract	iii
Σύνοψη	iv
Πίνακας Περιεχομένων	v
Πίνακας Εικόνων.....	viii
Πίνακας Πινάκων	x
1 Εισαγωγή.....	- 11 -
1.1 Σχετικά με την ανίχνευση πτώσης.....	- 11 -
1.2 Σκοπός της πτυχιακής εργασίας	- 12 -
1.3 Συνοπτική περιγραφή πτυχιακής εργασίας.....	- 12 -
2 Ανασκόπηση της σχετικής βιβλιογραφίας.....	- 13 -
2.1 Σχετικά με τις πτώσεις.....	- 13 -
2.1.1 Ορισμοί.....	- 13 -
2.1.2 Ανατομία μίας πτώσης	- 14 -
2.1.3 Συνέπειες των πτώσεων	- 14 -
2.2 Κατηγορίες συστημάτων ανίχνευσης πτώσης.....	- 15 -
2.2.1 Συσκευές στο περιβάλλον του χρήστη.....	- 15 -
2.2.2 Φορητές συσκευές	- 16 -
2.2.3 Ανίχνευση πτώσης με χρήση έξυπνων κινητών συσκευών	- 16 -
2.3 Αξιολόγηση συστημάτων ανίχνευσης πτώσης.....	- 17 -
2.4 Συστήματα ανίχνευσης πτώσης με χρήση αλγορίθμων βασισμένων σε κατώφλι	- 18 -
2.5 Συστήματα ανίχνευσης πτώσης με χρήση αλγορίθμων μηχανικής μάθησης.....	- 19 -
3 Δημιουργία του πειραματικού συνόλου δεδομένων	- 21 -
3.1 Λογισμικό και υλικό	- 21 -
3.2 Πειραματικά δεδομένα	- 22 -
3.2.1 Εθελοντές	- 22 -

3.2.2	Πτώσεις	- 23 -
3.2.3	Καθημερινές δραστηριότητες (ADLs)	- 24 -
3.3	Αρχεία.....	- 26 -
3.3.1	Αρχικά δεδομένα.....	- 26 -
3.3.2	Ονοματολογία αρχείων	- 27 -
4	Μεθοδολογία Υλοποίησης και Ανάπτυξη Εφαρμογών	- 29 -
4.1	Πλατφόρμες και εργαλεία που χρησιμοποιήθηκαν	- 29 -
4.1.1	Eclipse	- 29 -
4.1.2	Android.....	- 29 -
4.1.3	Matlab.....	- 32 -
4.1.4	WEKA.....	- 33 -
4.2	Ανάπτυξη εφαρμογής Android	- 33 -
4.2.1	Δημιουργία γραφικής διεπαφής	- 33 -
4.2.2	Συλλογή δεδομένων από αισθητήρες	- 35 -
4.2.3	Διαχείριση Εθελοντών και Δραστηριοτήτων	- 36 -
4.2.4	Εγγραφή δεδομένων σε αρχεία	- 39 -
4.2.5	Ρυθμίσεις Συστήματος	- 40 -
4.2.6	Παρουσίαση χρήσης εφαρμογής	- 42 -
4.3	Επεξεργασία δεδομένων	- 48 -
4.3.1	Εισαγωγή αρχείων στο Matlab.....	- 48 -
4.3.2	Συγχρονισμός αρχείων	- 48 -
4.3.3	Annotation.....	- 48 -
4.3.4	Τελικά δεδομένα	- 50 -
4.4	Αλγόριθμοι που βασίζονται σε κατοφλίωση	- 50 -
4.5	Αλγόριθμοι μηχανικής μάθησης.....	- 52 -
4.5.1	Εξαγωγή των χαρακτηριστικών	- 52 -
4.5.2	Χρήση αλγορίθμων μηχανικής μάθησης.....	- 53 -

5	Πειραματικά Αποτελέσματα.....	- 56 -
5.1	Αλγόριθμοι βασισμένοι σε κατώφλια.....	- 56 -
5.2	Αποτελέσματα ανίχνευσης πτώσης και κατηγοριοποίησης πτώσης με τεχνικές μηχανικής μάθησης	- 58 -
6	Επίλογος.....	- 62 -
6.1	Συμπεράσματα.....	- 62 -
6.2	Μελλοντικές Επεκτάσεις.....	- 62 -
	Βιβλιογραφία.....	- 64 -
	Παράρτημα Α- Ακρωνύμια.....	- 67 -

Πίνακας Εικόνων

Εικόνα 1: Κατευθύνσεις πτώσεων (Πηγή: [2])	- 14 -
Εικόνα 2: Φάσεις πτώσης (Πηγή: [2])	- 14 -
Εικόνα 3: Ανίχνευση πτώσεις μέσω ανάλυσης βίντεο(Πηγή: [4])	- 16 -
Εικόνα 4: Δημοσιευμένα άρθρα ανά χρονολογία (πηγή: [3]).....	- 17 -
Εικόνα 5: Πτώση στο πλάι	- 24 -
Εικόνα 6: Αρχείο με τα δεδομένα του επιταχυνσιόμετρου	- 27 -
Εικόνα 7: Αρχιτεκτονική Android	- 31 -
Εικόνα 8: Γραφική υλοποίηση προσθήκης χρήστη	- 34 -
Εικόνα 9: Δημιουργία της διάταξης (layout)για την αρχική οθόνη.....	- 35 -
Εικόνα 10: Μέθοδοι ενεργοποίησης και απενεργοποίησης αισθητήρων.....	- 36 -
Εικόνα 11: Δημιουργία κλάσης User	- 37 -
Εικόνα 12: Δημιουργία της κλάσης HumanActivity	- 37 -
Εικόνα 13: Δημιουργία κλάσης DatabaseHandler	- 38 -
Εικόνα 14: Αρχεία JSON με τους εθελοντές και της δραστηριότητες	- 39 -
Εικόνα 15: Υλοποίηση κλάσης WriteToFile	- 39 -
Εικόνα 16: Υλοποίηση της μεθόδου onSensorChanged()	- 40 -
Εικόνα 17: Δημιουργία xml αρχείου με τις ρυθμίσεις.....	- 41 -
Εικόνα 18: Ρυθμίσεις εφαρμογής.....	- 41 -
Εικόνα 19: Αρχική οθόνη εφαρμογής (β) οθόνη καταγραφής.....	- 42 -
Εικόνα 20: Επιλογή εθελοντή και δραστηριότητας πριν την καταγραφή.....	- 43 -
Εικόνα 21: Καταγραφή σε εξέλιξη.....	- 43 -
Εικόνα 22: Ολοκλήρωση καταγραφής.....	- 44 -
Εικόνα 23: Διαχείριση εθελοντών.....	- 45 -
Εικόνα 24: Επεξεργασία εθελοντή.....	- 46 -
Εικόνα 25: Διαγραφή εθελοντή.....	- 47 -
Εικόνα 26: Διαχείριση δραστηριοτήτων	- 47 -

Εικόνα 27: Annotation	- 49 -
Εικόνα 28: Τελική μορφοποίηση του αρχείου με δεδομένα επιταχυνσιόμετρου	- 50 -
Εικόνα 29: Σύγκριση του κάθε αλγόριθμου ανάλογα με τον τύπο της πτώσης.....	- 57 -
Εικόνα 30: Σύγκρισή των διαφορετικών σετ χαρακτηριστικών στην ανίχνευση πτώσης και ταξινόμηση πτώσης	- 61 -

Πίνακας Πινάκων

Πίνακας 1: Αποτελέσματα ενός συστήματος ανίχνευσης πτώσης.....	- 17 -
Πίνακας 2: Εθελοντές.....	- 23 -
Πίνακας 3: Τύποι πτώσεων	- 23 -
Πίνακας 4: Καθημερινές δραστηριότητες.....	- 25 -
Πίνακας 5: Αισθητήρες	- 28 -
Πίνακας 6: Χαρακτηριστικά και λειτουργίες του Android.....	- 30 -
Πίνακας 7: Παράμετροι που χρησιμοποιούνται από κάθε αλγόριθμο	- 51 -
Πίνακας 8: Αποτελέσματα αξιολόγησης threshold-based αλγορίθμων	- 57 -
Πίνακας 9: Αποτελέσματα ανίχνευσης και ταξινόμησης πτώσης με χρήση του συνόλου των χαρακτηριστικών	- 58 -
Πίνακας 10: Ανίχνευση και ταξινόμηση πτώσης με χρήση των χαρακτηριστικών που επέλεξε το BestFirst φίλτρο	- 59 -
Πίνακας 11: Ανίχνευση και ταξινόμηση πτώσης με χρήση των 10 επιλεγμένων χαρακτηριστικών.....	- 59 -
Πίνακας 12: Αποτελέσματα ανίχνευσης και ταξινόμησης πτώσης με χρήση μόνο του χαρακτηριστικού EEB.....	- 60 -

1 Εισαγωγή

1.1 Σχετικά με την ανίχνευση πτώσης

Ένα πρόβλημα που αντιμετωπίζουν οι άνθρωποι και κυρίως οι ηλικιωμένοι είναι οι πτώσεις. Πολλοί τραυματισμοί οφείλονται σε πτώσεις και αυτό μπορεί να προκαλέσει την διαμονή των ανθρώπων για μεγάλο χρονικό διάστημα στο νοσοκομείο. Ανάλογα με τον τύπο της πτώσης και την ηλικία του ατόμου υπάρχει μεγάλη πιθανότητα να έχουμε πτώσεις που να προκαλέσουν σοβαρές σωματικές βλάβες και ακόμη να αφήσουν αναίσθητο τον άνθρωπο στο πάτωμα. Μία πτώση μπορεί να συμβεί σε διάφορες στιγμές κατά τη διάρκεια της ημέρας και συνήθως κατά την διάρκεια κάποιας καθημερινής δραστηριότητας (Activity of Daily Living) (ADL). Υπάρχουν διάφορες προσεγγίσεις με τις οποίες μπορούμε να «παρακολουθήσουμε» ένα άτομο στο οποίο είναι πιθανό να συμβεί μία πτώση. Αυτές είναι:

- Χρήση κάμερας η οποία καταγράφει τις κινήσεις του ατόμου και με επεξεργασία εικόνας και κατάλληλους αλγορίθμους εντοπίζει την πτώση.
- Χρήση αισθητήρων στο περιβάλλον του ατόμου οι οποίοι ανιχνεύουν πτώσεις.(αισθητήρες πίεσης, υπέρυθροι αισθητήρες κ.α.)
- Τοποθέτηση συσκευών επάνω στο άτομο όπως επιταχυνσιόμετρα και κινητά τηλέφωνα. [1]

Μία συσκευή τοποθετημένη επάνω στο άτομο, σε σύγκριση με τις άλλες προσεγγίσεις έχει τα πλεονεκτήματα ότι μπορεί να χρησιμοποιηθεί από ανθρώπους επάνω ή μέσα από τα ρούχα τους χωρίς να χρειάζεται εγκατάσταση εξοπλισμού όπως γίνεται στις άλλες δύο προσεγγίσεις. Αυτό που θα πρέπει να κάνει μία συσκευή ανίχνευσης πτώσης είναι:

- Ανίχνευση της πτώσης με χρήση αλγόριθμων και
- επικοινωνία της συσκευής μετά την πτώση ώστε να ειδοποιήσει για βοήθεια σε περίπτωση που το άτομο δεν μπορεί να το κάνει ή έχει κάποιο σοβαρό τραυματισμό.

Η πλειοψηφία των “έξυπνων” κινητών τηλεφώνων (smartphones) έρχονται στην αγορά με ενσωματωμένους αισθητήρες οι οποίοι μπορούν να διαβάσουν συγκεκριμένες κινήσεις του τηλεφώνου όπως το επιταχυνσιόμετρο (accelerometer), το γυροσκόπιο (gyroscope), ο αισθητήρας προσανατολισμού (orientation sensor), κ.α. Έτσι λοιπόν με την χρήση των αισθητήρων αυτών και την μεγάλη υπολογιστική ικανότητα που έχουν πλέον τα έξυπνα κινητά τηλέφωνα, έχουμε την δυνατότητα να δημιουργήσουμε αλγόριθμους οι οποίοι θα μπορούν με επιτυχία να ανιχνεύσουν πτώσεις. Επίσης το κινητό τηλέφωνο προσφέρει την δυνατότητα της άμεσης επικοινωνίας με κλήσεις, γραπτά μηνύματα (SMS) και πρόσβαση στο διαδίκτυο. Έτσι μπορεί εύκολα να επικοινωνήσει και να καλέσει για βοήθεια εάν συμβεί μία πτώση. Η συνεχή πτώση των τιμών τους τα τελευταία χρόνια αλλά και το γεγονός ότι εύκολα ο κάθε άνθρωπος μπορεί να προμηθευτεί ένα σε σχέση με τις ειδικές συσκευές ανίχνευσης πτώσης το κάνει προσβάσιμο σε πολλούς ανθρώπους. Επίσης ένα άλλο πλεονέκτημα είναι ότι δεν χρειάζεται η κατασκευή εξειδικευμένου υλικού (hardware) αλλά μόνο η ανάπτυξη λογισμικού (software) που μπορεί να εκτελείται στο περιβάλλον ενός έξυπνου κινητού τηλεφώνου.

Στη παρούσα εργασία προσεγγίζεται το πρόβλημα της ανίχνευσης πτώσης με την χρήση έξυπνων κινητών τηλεφώνων. Αρχικά αναπτύχθηκε η εφαρμογή καταγραφής η οποία χρησιμοποιείται σε κινητές συσκευές Android και καταγράφει τα δεδομένα των αισθητήρων. Στη συνέχεια δημιουργήθηκε ένα σετ δεδομένων με διάφορους τύπους πτώσεων και

καθημερινών δραστηριοτήτων. Αυτό το σετ δεδομένων χρησιμοποιήθηκε για την αξιολόγηση τριών γνωστών αλγορίθμων που έχουν χρησιμοποιηθεί σε κινητά τηλέφωνα για την ανίχνευση πτώσης. Επίσης, σε δεύτερη φάση, με την χρήση τεχνικών μηχανικής μάθησης γίνεται ανίχνευση πτώσης αλλά και ταξινόμηση του είδους της κάθε πτώσης.

1.2 Σκοπός της πτυχιακής εργασίας

Σκοπός της πτυχιακής εργασίας είναι να προσφέρει ένα σετ δεδομένων που περιλαμβάνει δεδομένα τεσσάρων τύπων πτώσεων και καθημερινές δραστηριότητες από τους αισθητήρες επιτάχυνσης, γυροσκόπιου και προσανατολισμού του κινητού τηλεφώνου έτσι ώστε να υπάρχει ένα διαθέσιμο σετ δεδομένων για την ανάπτυξη αλγορίθμων σχετικά με την ανίχνευση πτώσης αλλά και την αναγνώριση κινήσεων.

1.3 Συνοπτική περιγραφή πτυχιακής εργασίας

Στο πρώτο κεφάλαιο γίνεται μία περίληψη της εργασίας, και στο δεύτερο παρουσιάζονται τα χαρακτηριστικά των πτώσεων, τα συστήματα ανίχνευσης πτώσης που ήδη υπάρχουν και οι κατηγορίες τους.

Στο τρίτο κεφάλαιο περιγράφεται η διαδικασία σύνθεσης του πειραματικού συνόλου δεδομένων (experimental dataset). Στο τέταρτο κεφάλαιο περιγράφονται τα εργαλεία και οι πλατφόρμες που χρησιμοποιήθηκαν καθώς και η ανάπτυξη της εφαρμογής καταγραφής και των αλγορίθμων.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα αποτελέσματα των αλγορίθμων και γίνεται η ανάλυσή τους. Τέλος στο έκτο κεφάλαιο αναφέρονται τα συμπεράσματα και προτάσεις για μελλοντική επέκταση της εργασίας.

2 Ανασκόπηση της σχετικής βιβλιογραφίας

Η ανίχνευση πτώσης είναι ένα ζήτημα που απασχολεί την παγκόσμια ερευνητική κοινότητα. Κάθε χρόνο όλο και περισσότερα ερευνητικά άρθρα δημοσιεύονται σχετικά με την ανίχνευση πτώσης. Σε αυτό το κεφάλαιο παρουσιάζεται λεπτομερώς το πρόβλημα των πτώσεων, τί είναι, πώς συμβαίνουν και γίνεται αναφορά στις διαφορετικές προσεγγίσεις του προβλήματος επικεντρώνοντας στα συστήματα ανίχνευσης πτώσης σε έξυπνα κινητά τηλέφωνα.

2.1 Σχετικά με τις πτώσεις

Το πρόβλημα των πτώσεων στους ηλικιωμένους ανθρώπους έχει μεγάλες οικονομικές και κοινωνικές επιδράσεις. Οι πτώσεις είναι οι βασική αιτία που οι ηλικιωμένοι παραμένουν στο νοσοκομείο για μεγάλο χρονικό διάστημα και είναι η βασική αιτία θανάτου σε ηλικίες άνω των 75 ενώ στις ηλικίες μεταξύ 65 - 75 είναι η δεύτερη κατά σειρά αιτία θανάτου. Στους ανθρώπους που πάσχουν από Alzheimer η πιθανότητα πτώσης τριπλασιάζεται [2].

Κάθε χρόνο 11 εκατομμύρια άνθρωποι πέφτουν. Το 2005 έγιναν περίπου 56,423 εισαγωγές σε νοσοκομείο από πτώσεις και 7,946 θάνατοι στις Ηνωμένες Πολιτείες οι οποίοι προκλήθηκαν από πτώσεις. Πολλοί από αυτούς τους θανάτους είναι αποτέλεσμα της «ακινησίας για μεγάλο χρονικό διάστημα» (long-lie), δηλαδή της μεγάλης διάρκειας που ο άνθρωπος βρίσκεται στο πάτωμα ανήμπορος να καλέσει για βοήθεια. Επίσης ο φόβος από το long-lie μπορεί να μειώσει την ποιότητα ζωής των ηλικιωμένων [1].

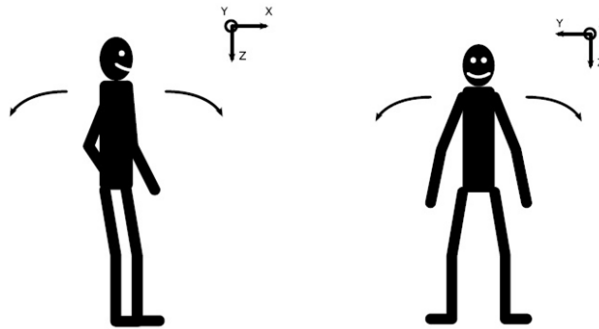
Τέλος, σύμφωνα με τους Igual et al. [3] εάν δεν ληφθούν προληπτικά μέτρα στο άμεσο μέλλον, ο αριθμός των τραυματισμών θα αυξηθεί κατά 100% το έτος 2030. Επομένως είναι αναγκαία η ανάπτυξη βοηθητικών συσκευών που θα μπορούσαν να ανακουφίσουν από αυτό το πρόβλημα.

Λαμβάνοντας υπόψη τα παραπάνω συμπεραίνουμε ότι το πρόβλημα των πτώσεων είναι ένα σημαντικό πρόβλημα που συμβαίνει τακτικά σε ηλικιωμένους ανθρώπους, μειώνει την ποιότητα ζωής τους, έχει τεράστιες οικονομικές επιπτώσεις και αυξάνεται με μεγάλους ρυθμούς.

2.1.1 Ορισμοί

Μία πτώση μπορεί να οριστεί με διαφορετικούς τρόπους ανάλογα το τρόπο που την μελετάμε. Στην παρούσα μελέτη εστιάζουμε στην ανάλυση των ανθρώπινων κινήσεων. Ένας ορισμός της πτώσης είναι: «Ακούσιος προορισμός προς το έδαφος ή σε ένα χαμηλότερο επίπεδο που έχει ως συνέπεια ένα βίαιο χτύπημα, απώλεια των αισθήσεων και ξαφνική παράλυση» [2].

Από την άποψη της ανθρώπινης ανατομίας, μια πτώση συνήθως συμβαίνει σε 2 κατευθύνσεις. Η πρώτη είναι εμπρός ή πίσω και η δεύτερη είναι στο πλάι (Εικόνα 1).

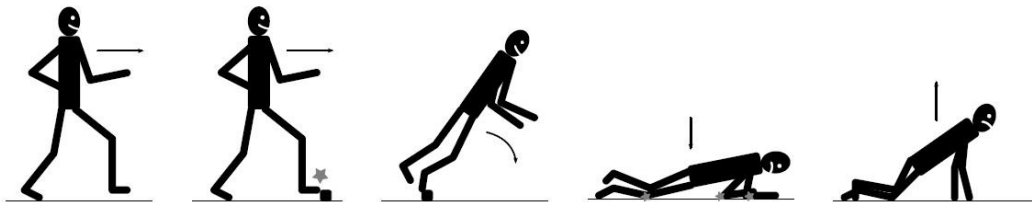


Εικόνα 1: Κατευθύνσεις πτώσεων (Πηγή: [2])

2.1.2 Ανατομία μίας πτώσης

Μία πτώση γενικά είναι συνέπεια μίας καθημερινής δραστηριότητας και ξεκινάει από ένα απρόβλεπτο συμβάν όπως να σκοντάψουμε, να γλιστρήσουμε ή να χάσουμε την ισορροπία μας. Στη συνέχεια ακολουθεί η ελεύθερη πτώση και η σύγκρουση με το πάτωμα. Συνήθως το άτομο μετά την πτώση παραμένει στο πάτωμα για μερικά δευτερόλεπτα ή για ώρες, ανάλογα με το πόσο σοβαρός θα είναι ο τραυματισμός, και στη συνέχεια προσπαθεί να σηκωθεί από μόνο του ή με την βοήθεια κάποιου άλλου ανθρώπου. Οι πέντε φάσεις μίας πτώσης φαίνονται στην Εικόνα 2.

1. Καθημερινή δραστηριότητα
2. Απρόβλεπτο συμβάν
3. Ελεύθερη πτώση
4. Σύγκρουση
5. Ανάκτηση (Προαιρετικό)



Εικόνα 2: Φάσεις πτώσης (Πηγή: [2])

2.1.3 ΣΥΝΕΠΕΙΕΣ ΤΩΝ ΠΤΩΣΕΩΝ

Οι πτώσεις είναι η κύρια αιτία εισαγωγής σε νοσοκομείο και η έκτη αιτία θανάτου στις ηλικίες πάνω των 65. Σε ηλικίες μεταξύ 65-75 είναι η δεύτερη αιτία θανάτου ενώ η πρώτη σε ηλικίες πάνω των 75 [2].

2.1.3.1 Σωματικές βλάβες

Οι γρατζουνιές και οι μώλωπες είναι τα ελαφριά τραύματα μίας πτώσης. Σε χειρότερες περιπτώσεις οι τραυματισμοί είναι στο κάτω μέρος του σώματος, κυρίως στο ισχίο. Στο πάνω μέρος του σώματος στο κεφάλι και στον κορμό τα τραύματα είναι τα πιο συχνά. Περίπου 66% των εισαγωγών σε νοσοκομεία οφείλονται σε τουλάχιστον ένα κατάγμα [2].

2.1.3.2 Ψυχολογικές βλάβες

Μία πτώση επηρεάζει την αυτοπεποίθηση του ατόμου. Συνέπειες είναι ο φόβος, η απώλεια ανεξαρτησίας, οι περιορισμένες δυνατότητες, η χαμηλή αυτοεκτίμηση και γενικά η χαμηλή ποιότητα ζωής [2].

2.1.3.3 Οικονομικές συνέπειες

Οι οικονομικές δαπάνες χωρίζονται σε άμεσες και έμμεσες. Οι άμεσες δαπάνες που σχετίζονται με τις πτώσεις οφείλονται σε ιατρικές εξετάσεις, σε θεραπείες αποκατάστασης, σε αναπηρικά καροτσάκια, μαστούνια, κλπ. Οι έμμεσες δαπάνες είναι από τις συνέπειες του θανάτου των ασθενών. Έχει διαπιστωθεί από μελέτες ότι το έτος 2.000 οι δαπάνες από γεγονότα συσχετιζόμενα με πτώσεις ήταν πάνω από 19 εκατομμύρια δολάρια ενώ εκτιμάται ότι το έτος 2020 θα φθάσουν τα 54,9 εκατομμύρια δολάρια. Αυτό δείχνει ότι χρόνο με το χρόνο τα έξοδα στην υγεία από τις πτώσεις αυξάνονται δραματικά [2].

2.2 Κατηγορίες συστημάτων ανίχνευσης πτώσης

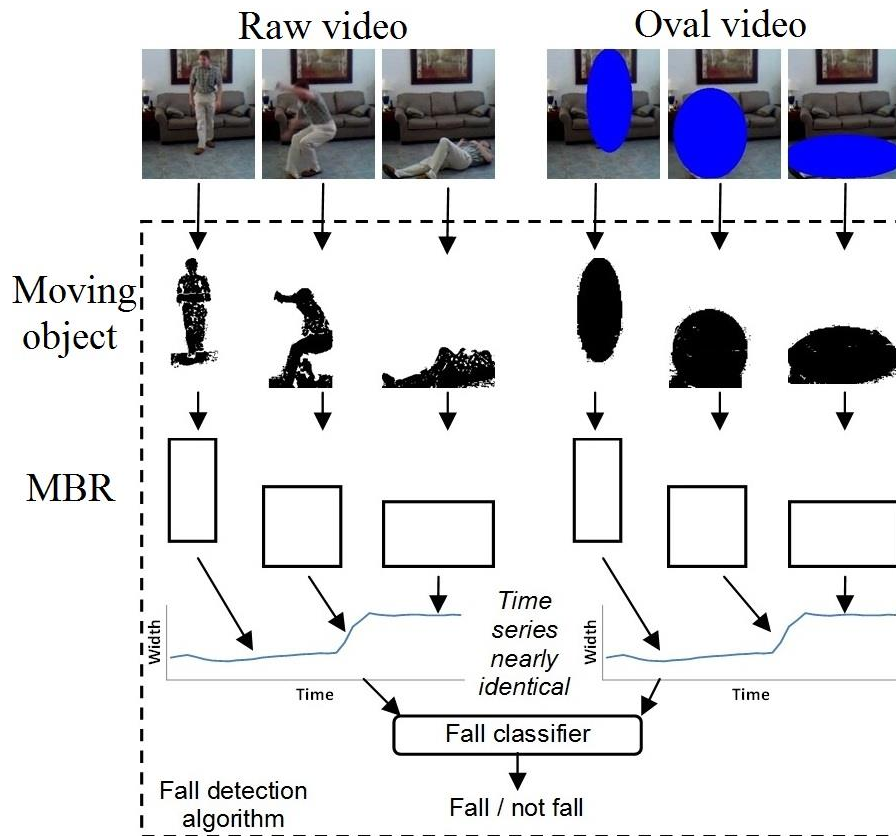
Το πρόβλημα της ανίχνευσης πτώσης έχει μελετηθεί πολύ και υπάρχουν διάφορα συστήματα που προσπαθούν και έχουν ως στόχο την επίλυση του προβλήματος αυτού. Υπάρχουν δύο διαφορετικές κατηγορίες χρησιμοποιώντας διαφορετικές τεχνολογίες και προσεγγίζοντας το πρόβλημα από διαφορετικές σκοπιές. Οι κατηγορίες που αναλύονται παρακάτω και είναι:

- Συσκευές στο περιβάλλον του χρήστη (context-aware systems).
- Φορετές συσκευές (wearable devices).

2.2.1 Συσκευές στο περιβάλλον του χρήστη

Αυτά τα συστήματα χρησιμοποιούν αισθητήρες στο περιβάλλον του χρήστη για να ανιχνεύσουν πτώσεις. Το μεγάλο τους πλεονέκτημα είναι ότι το άτομο δεν χρειάζεται να φορέσει κάποια ειδική συσκευή. Οι πιο συνηθισμένοι τύποι αισθητήρων είναι κάμερες, αισθητήρες πατώματος, υπέρυθροι αισθητήρες, μικρόφωνα και αισθητήρες πίεσης.

Τα συστήματα που χρησιμοποιούν κάμερες χρησιμοποιούν τεχνικές τεχνητής όρασης για την ανίχνευση πτώσης. Οι πλειοψηφία αυτών των συστημάτων ξεκινούν με την εξαγωγή των χαρακτηριστικών, για παράδειγμα μερικά από αυτά μπορεί να είναι, το ύψος και το βάρος των ανθρώπων, οι ακμές στο περίγραμμα του σώματος, οι αλλαγές στην φωτεινότητα, ο προσανατολισμός του κύριου άξονα του ανθρώπου, κλπ. Στην Εικόνα 3 φαίνεται η εξαγωγή του χαρακτηριστικού MBR (Minimum Bounding Rectangle) για την μέθοδο που βασίζεται σε επεξεργασία βίντεο, το οποίο είναι το ελάχιστο παραλληλόγραμμο της σιλουέτας, και η εναλλαγή του πλάτους του καθώς γίνεται η πτώση. Στη συνέχεια αυτά τα χαρακτηριστικά συγκρίνονται και περνάνε μέσα από αλγόριθμους ταξινόμησης για να ξεχωρίσουν ADL από πτώσεις χρησιμοποιώντας διάφορες τεχνικές. Παρόλο που οι περισσότερες μελέτες έχουν γενικά υψηλή ακρίβεια στα αποτελέσματά τους, αυτές οι μελέτες έχουν γίνει σε ελεγχόμενο περιβάλλον και ίσως μία μελέτη σε βάθος χρόνου να δώσει χαμηλότερα αποτελέσματα. Τα μειονεκτήματα αυτής της προσέγγισης είναι η έλλειψη ηλικιωμένων εθελοντών αλλά και το ότι οι άνθρωποι δεν δέχονται να εγκαταστήσουν κάμερες στο χώρο τους [3].



Εικόνα 3: Ανίχνευση πτώσεις μέσω ανάλυσης βίντεο (Πηγή: [4])

2.2.2 Φορητές συσκευές

Οι φορητές συσκευές (wearable devices) είναι μικρές ηλεκτρονικές συσκευές εξοπλισμένες με αισθητήρες και οι οποίες φοριούνται από τους χρήστες μέσα ή κάτω από τα ρούχα. Η πλειοψηφία αυτών των συσκευών περιέχουν τον αισθητήρα επιτάχυνσης. Κάποιες από αυτές ίσως να περιέχουν και γυροσκόπια ώστε να μπορούν να παίρνουν πληροφορίες για την θέση του ανθρώπου. Η χρήση τους είναι στην αξιολόγηση του βήματος και ισορροπίας, στο ρίσκο της πτώσης και στην παρακολούθηση ασθενών. Αυτή η τάση έχει αυξηθεί τα τελευταία χρόνια αφού οι αισθητήρες αυτοί είναι φθηνοί και διαθέσιμοι, όπως επίσης και τα έξυπνα κινητά τηλέφωνα που έχουν ενσωματωμένους αυτούς τους αισθητήρες.

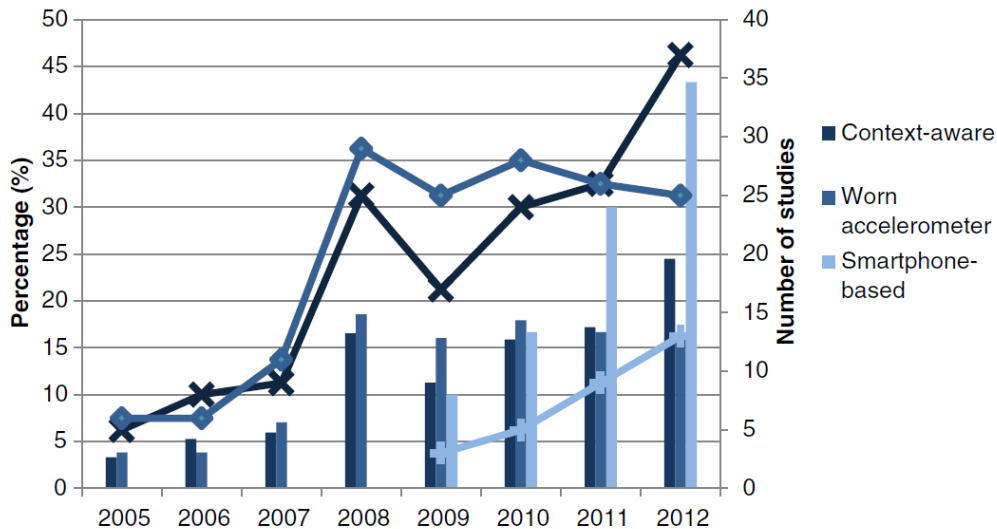
Αυτή η προσέγγιση έχει πλεονεκτήματα όπως χαμηλό κόστος εγκατάστασης (εσωτερικοί και εξωτερικοί χώροι) και μικρό μέγεθος. Μειονεκτήματα των συσκευών αυτών είναι ότι οι ηλικιωμένοι συχνά αρνούνται ή ξεχνάνε να φορέσουν τις συσκευές ή ακόμα ξεχνάνε να φορτίσουν τις μπαταρίες [3].

2.2.3 Ανίχνευση πτώσης με χρήση έξυπνων κινητών συσκευών

Με την συνεχή βελτίωση της τεχνολογίας στα κινητά τηλέφωνα ολοένα αυξάνεται η υπολογιστική ισχύς των κινητών τηλεφώνων ενώ μειώνεται το κόστος τους. Για αυτό το λόγο όλο και περισσότεροι άνθρωποι χρησιμοποιούν έξυπνα κινητά τηλέφωνα. Πολλά από αυτά τα τηλέφωνα έχουν ενσωματωμένους αισθητήρες όπως το επιταχυνσιόμετρο, η ψηφιακή πυξίδα, το γυροσκόπιο, GPS, μικρόφωνο, και κάμερα δίνοντας την δυνατότητα σε πολλούς ερευνητές να αναπτύξουν συστήματα ανίχνευσης με τη χρήση έξυπνων κινητών τηλεφώνων [3]. Η

πλειοψηφία των κινητών αυτών συσκευών δίνει τη δυνατότητα στους προγραμματιστές να μπορούν να έχουν πρόσβαση στα σήματα των αισθητήρων. Έτσι είναι δυνατό να μελετηθεί το πρόβλημα της ανίχνευσης πτώσης.

Έτσι λαμβάνοντας υπόψη τα πλεονεκτήματα των κινητών τηλεφώνων και σε συνδυασμό με την συνεχή βελτίωση της τεχνολογίας, η ανίχνευση πτώσης μέσω αυτών ερευνάται όλο και περισσότερο.



Εικόνα 4: Δημοσιευμένα άρθρα ανά χρονολογία (πηγή: [3])

Στην Εικόνα 4 παρουσιάζεται ο αριθμός των δημοσιευμένων άρθρων σχετικά με την ανίχνευση πτώσης. Από το 2009 και έπειτα παρατηρείται μία ταχύτατη αύξηση των άρθρων που ασχολούνται με ανίχνευση πτώσης σε έξυπνα κινητά τηλέφωνα.

2.3 Αξιολόγηση συστημάτων ανίχνευσης πτώσης

Ένα σύστημα που ανιχνεύει πτώσεις θα πρέπει να είναι όσο το δυνατό περισσότερο ακριβές και αποτελεσματικό. Η ποιότητα ενός συστήματος ανίχνευσης πτώσης δίνεται από τους τρεις δείκτες με βάση τις τέσσερις πιθανές περιπτώσεις που φαίνονται στον Πίνακα 1.

	Έγινε πτώση	Δεν έγινε πτώση
Ανιχνεύθηκε	True Positive (TP)	False Positive (FP)
Δεν ανιχνεύθηκε	False Negative (FN)	True Negative (TN)

Πίνακας 1: Αποτελέσματα ενός συστήματος ανίχνευσης πτώσης

Εναισθησία (sensitivity) ή TP rate: Είναι η ικανότητα το σύστημα να ανιχνεύσει μία πτώση και υπολογίζεται από τον λόγο των ανιχνευμένων πτώσεων προς τις συνολικές πτώσεις.

$$\text{Sensitivity} = \text{TP rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Ειδικότητα (specificity): Είναι η ικανότητα του συστήματος να ανιχνεύει μία πτώση μόνο αν αυτή έχει συμβεί. Επίσης από το specificity μπορεί να υπολογιστεί και το FP rate.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{FP rate} = 1 - \text{specificity}$$

Ακρίβεια (accuracy): Είναι η ικανότητα του συστήματος να ανιχνεύει σωστά τόσο αν έγινε τόσο και αν δεν έγινε κάποια πτώση.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

2.4 Συστήματα ανίχνευσης πτώσης με χρήση αλγορίθμων βασισμένων σε κατώφλι

Τα συστήματα που χρησιμοποιούν αλγόριθμους βασισμένους σε κατώφλια για την ανίχνευση πτώσης (threshold-based fall detection algorithms) χρησιμοποιούν τα δεδομένα του επιταχυνσιόμετρου και/ή του προσανατολισμού για να ανιχνεύσουν τις διαφορετικές φάσεις μία πτώσης (βλ. 2.1.2).

Χρησιμοποιώντας τον αισθητήρα του επιταχυνσιόμετρου του κινητού τηλεφώνου οι *Sposaro και Tyson* [1] ανέπτυξαν μία εφαρμογή Android, την “iFall” η οποία ανιχνεύει πτώσεις όταν δύο όρια, ένα ανώτερο και ένα κατώτερο ξεπεραστούν μέσα σε ένα χρονικό διάστημα. Επίσης τα όρια μπορούν να παραμετροποιηθούν σύμφωνα με την ηλικία, το βάρος και το ύψος του χρήστη. Οι *Lee και Carlisle* [5] παρουσίασαν μία παρόμοια τεχνική στην οποία τα κατώφλια ορίζονται ξεχωριστά για κάθε χρήστη και βασίζονται στις μικρότερες και μεγαλύτερες τιμές που πάρθηκαν από τα ADLs. Επιπλέον, έκαναν μία σύγκριση μεταξύ των σημάτων του επιταχυνσιόμετρου του κινητού και ενός αισθητήρα στο σώμα, καταλήγοντας ότι το κινητό τηλέφωνο είναι κατάλληλο για ανίχνευση πτώσης και λειτουργεί το ίδιο καλά όπως και ο αισθητήρας του επιταχυνσιόμετρου. Οι *Dai et al.* [6] ανέπτυξαν μία εφαρμογή που ονομάζεται “PerFallD” η οποία λαμβάνει υπόψη της την συνολική επιτάχυνση του επιταχυνσιόμετρου και την απόλυτη τιμή του προσανατολισμού (orientation) και ελέγχει την διαφορά μεταξύ του μέγιστου και του ελάχιστου από κάθε ένα από τα σήματα εάν έχει περάσει ένα όριο μέσα σε δύο χρονικά διαστήματα. Οι τιμές των ορίων πάρθηκαν από καμπύλες αποτύπωσης λειτουργικών χαρακτηριστικών δέκτη (receiver operating characteristics - ROC) που έγιναν από τα δεδομένα εκπαίδευσης του αλγορίθμου. Η σύγκριση που έγινε με το κινητό τηλέφωνο τοποθετημένο σε διαφορετικά σημεία, όπως το στήθος και τη μέση, έδειξε ότι όταν το κινητό είναι τοποθετημένο στη μέση του ατόμου παράγονται τα καλύτερα αποτελέσματα. Οι *Tolkiehn et al.* [7] χρησιμοποίησαν έναν αισθητήρα επιτάχυνσης και έναν αισθητήρα βαρομετρικής πίεσης για να ανιχνεύσουν πτώσεις και την κατεύθυνσή τους. Η ανίχνευση της πτώσης περιλάμβανε δύο στάδια με όρια. Συγκεκριμένα η ξαφνική μεταβολή του προσανατολισμού μπορεί να οριστεί σαν πτώση και β) οι κορυφές της επιτάχυνσης ή ο συνδυασμός της γωνίας και της κρούσης μπορεί επίσης να οριστεί ως πτώση. Οι *He et al.* [8] χρησιμοποίησαν έναν αλγόριθμό ο οποίος πρώτα υπολογίζει την περιοχή που καλύπτεται από το μέγεθος των τριών αξόνων του επιταχυνσιόμετρου (signal magnitude area -SMA) για να βρει πόσο βίαιη είναι η δραστηριότητα του χρήστη. Στη συνέχεια υπολογίζεται

η συνολική επιτάχυνση και τέλος η γωνία κλίσης (tilt angle). Ο αλγόριθμος αυτός μπορεί να κατηγοριοποιήσει δραστηριότητες όπως: πτώση, οριζόντια δραστηριότητα, κάθετη δραστηριότητα και ακινησία χρησιμοποιώντας όρια στις τρεις υπολογισμένες τιμές. Οι *Fang et al.* [9] χρησιμοποίησαν τα δεδομένα της επιτάχυνσης και του προσανατολισμού από τους αισθητήρες ενός κινητού τηλεφώνου. Αν η διαφορά της μέγιστης και της ελάχιστης συνολικής επιτάχυνσης είναι μεγαλύτερη από ένα προκαθορισμένο όριο και η κάθετη επιτάχυνση είναι μικρότερη από ένα άλλο όριο στην ίδια χρονική περίοδο, τότε μία πτώση έχει συμβεί. Επιπλέον μία σύγκριση σε διαφορετικά μέρη που μπορεί να χρησιμοποιηθεί το κινητό τηλέφωνο έδειξε ότι στο στήθος ο αλγόριθμος αυτός αποδίδει καλύτερα. Μία παρόμοια προσέγγιση αναπτύχθηκε από τους *Cao et al.* [10] με την διαφορά ότι η παραμετροποίηση των ορίων γινόταν δυναμικά από το ύψος, το βάρος, την ηλικία και φύλο των χρηστών. Τέλος, οι *Viet et al.* [11] καθόρισαν το μέγιστο και ελάχιστο όριο υπολογίζοντας την μέγιστη και ελάχιστη κορυφή στην επιτάχυνση αντίστοιχα.

2.5 Συστήματα ανίχνευσης πτώσης με χρήση αλγορίθμων μηχανικής μάθησης

Η μηχανική μάθηση χαρακτηρίζεται από δύο φάσεις, η πρώτη αφορά την εκπαίδευση κάποιου αλγόριθμου και η δεύτερη την εφαρμογή του αλγορίθμου αυτού για κατηγοριοποίηση πειραματικών δεδομένων. Η εκπαίδευση του αλγορίθμου απαιτεί την συλλογή των δεδομένων, την εξαγωγή των χαρακτηριστικών από τα δεδομένα και τέλος την εκπαίδευση του αλγορίθμου. Ο έλεγχος περιλαμβάνει την κατασκευή του συστήματος και την παραγωγή των αποτελεσμάτων.

Οι *Zhang et al.* [12] πρότειναν μία μέθοδο η οποία σε πρώτη φάση ένας αλγόριθμος ταξινόμησης SVM χρησιμοποιείται για να πάρει τα δεδομένα που έχουν μεγάλη πιθανότητα να είναι πτώση, όμως δεν μπορούσαν να βρουν την πτώση με βεβαιότητα. Στο δεύτερο επίπεδο αυτά τα δεδομένα ταξινομούνται (classified) από το KFD (Kernel Fisher Discriminant) και K-NN (K nearest neighbor) αλγόριθμο. Για την εκπαίδευση (train) του αλγορίθμου χρησιμοποίησαν τα 2/3 των δεδομένων, τα οποία επιλέχθηκαν με τυχαίο τρόπο και το υπόλοιπο 1/3 χρησιμοποιήθηκε για τον έλεγχο(test).

Οι *Lustrek και Kaluza* [13] χρησιμοποίησαν οκτώ μεθόδους μηχανικής μάθησης. Τους *C.45 Decision Trees*, *RIPPER decision rules*, *Naïve Bayes*, *3-NN*, *SVM*, *RandomForest*, *Bagging*, *Adaboost M1 boosting* για την ανίχνευση πτώσης και την αναγνώριση κινήσεων. Ο αλγόριθμος *SVM* έδειξε ότι αποδίδει καλύτερα. Οι *Abbate et al.* [14] υλοποίησαν ένα σύστημα ανίχνευσης πτώσης στο οποίο τα δεδομένα από το επιταχυνσιόμετρο δόθηκαν σαν είσοδο σε ένα νευρωνικό δίκτυο. Τα αποτελέσματά τους είχαν 100% ακρίβεια. Οι *Albert et al.* [15] χρησιμοποίησαν τεχνικές μηχανικής μάθησης για την ανίχνευση πτώσης και την κατηγοριοποίηση της πτώσης χρησιμοποιώντας δεδομένα επιτάχυνσης από κινητό τηλέφωνο σε προκαθορισμένο σημείο στο σώμα με ειδική ζώνη. Τα χαρακτηριστικά που παρήχθησαν ήταν από το πεδίο του χρόνου και των συχνοτήτων. Πέντε αλγόριθμοι χρησιμοποιήθηκαν, συγκεκριμένα οι *SVM*, *Regularized Logistic Regression (RLR)*, *Naïve Bayes*, *k-NN* και *Decision Trees* χρησιμοποιήθηκαν με 10-fold cross-validation και subject-wise cross-validation. Το cross-validation είναι μία τεχνική επικύρωσης ενός μοντέλου με στόχο την αποτίμηση του βαθμού γενίκευσης των αποτελεσμάτων στατιστικής ανάλυσης ενός ανεξάρτητου συνόλου δεδομένων. Χρησιμοποιείται κυρίως σε χώρους όπου ο στόχος είναι η πρόβλεψη, και κάποιος θέλει να εκτιμηθεί με ακρίβεια το πώς ένα προγνωστικό μοντέλο θα λειτουργήσει στην πράξη.

Ένα άλλο σύστημα ανίχνευσης πτώσης αναπτύχθηκε από τους *Zhao et al.* [16] με όνομα “*FallAlarm*”. Σε αυτή τη μέθοδο τα σήματα της επιτάχυνσης αξιολογήθηκαν από ένα decision tree και σε σύγκριση με τους αλγόριθμους *Naïve Bayes* και *SVM* είχαν καλύτερα αποτελέσματα. Τα χαρακτηριστικά από το σήμα ήταν η διασπορά (standard deviation) και η κλίση (slope) για το πεδίο του χρόνου και για το πεδίο των συχνοτήτων η ενέργεια του σήματος και correlation. Τα χαρακτηριστικά πάρθηκαν από ένα μετακινούμενο παράθυρο με 50% επικάλυψη και μέγεθος 4 δευτερόλεπτα. Το σετ δεδομένων χωρίστηκε σε δύο υποσύνολα, το 50% αποτέλεσε το σύνολο εκμάθησης (training set) και το άλλο 50% αποτέλεσε το σύνολο δοκιμής του αλγορίθμου προς κατηγοριοποίηση (test set). Οι *Fahmi et al.* [17] πρότειναν έναν αλγόριθμο ο οποίος βασίζεται σε ένα δένδρο αποφάσεων (decision tree) και τα χαρακτηριστικά της κάθε πτώσης. Τέλος οι *Kansiz et al.* [18] ανέπτυξαν ένα σύστημα βασισμένο σε χαρακτηριστικά στο πεδίο του χρόνου με συχνότητα δειγματοληψίας 20 Hz. Για την εξαγωγή των χαρακτηριστικών χρησιμοποιήθηκε ένα χρονικό παράθυρο διάρκειας τριών δευτερολέπτων και για την κατηγοριοποίηση επιλέχθηκαν οι αλγόριθμοι: *K-Star*, *Decision Tree* και *NaiveBayes*. Ο αλγόριθμος *K-Star* είχε καλύτερα αποτελέσματα από τους άλλους με τη μέθοδο 10-fold cross validation.

3 Δημιουργία του πειραματικού συνόλου δεδομένων

Η έλλειψη ενός δημόσιου και διαθέσιμου συνόλου δεδομένων που θα περιλαμβάνει διαφορετικούς τύπους πτώσεων δημιούργησε την ανάγκη για την δημιουργία ενός τέτοιου συνόλου δεδομένων. Για την δημιουργία του χρησιμοποιήθηκαν οι αισθητήρες από ένα κινητό τηλέφωνο τελευταίας τεχνολογίας. Το συγκεκριμένο σετ δεδομένων μπορεί να χρησιμοποιηθεί όχι μόνο για την σύγκριση διαφόρων αλγορίθμων ανίχνευσης πτώσης αλλά και για αναγνώριση κινήσεων αφού περιλαμβάνει αρκετές καθημερινές δραστηριότητες. Για λόγους ευχρηστίας, οι προσομοιώσεις έγιναν με όσο το δυνατό πιο ρεαλιστικό τρόπο όσον αφορά την θέση του κινητού τηλεφώνου επάνω στον εθελοντή.

Οι τύποι των πτώσεων που επιλέχθηκαν είναι οι πιο κοινές πτώσεις που μπορούν να συμβούν και αναλύουν οι Bourke et al. [19]. Οι καθημερινές δραστηριότητες επιλέχθηκαν με τρία κριτήρια. Το πρώτο είναι οι δραστηριότητες που μοιάζουν με πτώσεις καθώς υπάρχει μεταβολή από μία συγκεκριμένη κατάσταση σε μία άλλη, που πολλές φορές καταλήγει σε ακινησία, όπως οι δραστηριότητες: «κάθισμα σε καρέκλα», «έξοδος από το αυτοκίνητο», «είσοδος σε αυτοκίνητο». Το δεύτερο κριτήριο ήταν οι βίαιες κινήσεις καθώς ή συμπεριφορά τους επίσης μοιάζει με αυτή των πτώσεων αφού αποτελούνται και αυτές από απότομες κινήσεις. Τέτοιες κινήσεις είναι τα άλματα και το τρέξιμο. Το τρίτο κριτήριο ήταν οι πιο συχνές καθημερινές κινήσεις. Αυτές οι κινήσεις ενδέχεται να μην επηρεάσουν τους αλγόριθμους ανίχνευσης πτώσης όμως θεωρήθηκε αναγκαίο να συμπεριληφθούν καθώς είναι οι πιο συχνές κινήσεις που γίνονται κατά την διάρκεια της ημέρας από τους ανθρώπους. Επίσης έχοντας υπόψη και το πρόβλημα της αναγνώρισης κινήσεων, οι συγκεκριμένες κινήσεις δίνουν την δυνατότητα να ερευνηθεί και αυτός ο τομέας από το συγκεκριμένο σετ δεδομένων. Τέτοιες κινήσεις είναι το περπάτημα, η στάση σε όρθια θέση και τα ανέβασμα και κατέβασμα σκαλοπατιών.

3.1 Λογισμικό και υλικό

Το λογισμικό που αναπτύχθηκε και χρησιμοποιήθηκε για την δημιουργία του σετ δεδομένων είναι μία εφαρμογή για κινητές συσκευές Android που περιλαμβάνουν τους αισθητήρες επιταχυνσιόμετρο, γυροσκόπιου και προσανατολισμού, καθώς και αποθηκευτικό χώρο. Η εφαρμογή αυτή μπορεί να χρησιμοποιηθεί σε οποιαδήποτε συσκευή με λειτουργικό σύστημα Android version 2.1 και νεότερη και περιλαμβάνει αυτούς τους αισθητήρες. Η λεπτομερής ανάλυση της εφαρμογής γίνεται στο υποκεφάλαιο 4.2. Συνοπτικά, η εφαρμογή αυτή χρησιμοποιεί μία βάση δεδομένων για να αποθηκεύει τους εθελοντές, τις πτώσεις και τις δραστηριότητες. Ο χρήστης της εφαρμογής επιλέγει από το μενού τις διάφορες παραμέτρους που επιθυμεί και στη συνέχεια ξεκινάει την καταγραφή. Όταν η καταγραφή τελειώσει δημιουργούνται τα αρχεία που περιέχουν τα δεδομένα που κατέγραψαν οι αισθητήρες του κινητού τηλεφώνου και τα αποθηκεύει στην κάρτα μνήμης της συσκευής.

Το κινητό τηλέφωνο που επιλέχθηκε για τις καταγραφές είναι τελευταίας τεχνολογίας κινητό τηλέφωνο. Συγκεκριμένα το Samsung Galaxy S III (gt-I-9300) [20] με τους αισθητήρες επιταχυνσιόμετρο και γυροσκοπίου στην ενσωματωμένη μονάδα LSM330DLC [21] κατασκευασμένη από την εταιρία STMicroelectronics. Επίσης ο υπολογισμός του προσανατολισμού γίνεται με υπολογιστικό τρόπο (software-based) και υπολογίζεται, σύμφωνα με το Android API [22], από τα δεδομένα των αισθητήρων της επιτάχυνσης και του γεωμαγνητικού πεδίου (geomagnetic field sensor).

3.2 Πειραματικά δεδομένα

Οι εθελοντές πραγματοποίησαν τις δραστηριότητες και τις πτώσεις. Για να γίνει προσομοίωση της καθημερινής χρήσης του κινητού τηλεφώνου όλοι οι εθελοντές τοποθέτησαν το κινητό τηλέφωνο σε μία τσέπη στο παντελόνι την οποία οι ίδιοι επέλεξαν και με τυχαίο προσανατολισμό. Αυτό έγινε ώστε να είναι όσο πιο ρεαλιστικός ο τρόπος με τον οποίον θα καταγραφούν τα δεδομένα προσομοιώνοντας τον πραγματικό κόσμο. Όλες οι δραστηριότητες έγιναν σε χώρο του ΤΕΙ Κρήτης. Οι πτώσεις έγιναν στο χώρο του κλειστού γυμναστηρίου του ΤΕΙ Κρήτης με την επίβλεψη του συγγραφέα ώστε να είναι σίγουρο ότι οι πτώσεις έγιναν με τον σωστό τρόπο. Για τις πτώσεις χρησιμοποιήθηκαν τρία στρώματα πολεμικών τεχνών πάχους 5 εκ. τα οποία ήταν σχετικά σκληρά και τοποθετήθηκαν το ένα δίπλα στο άλλο. Το κινητό τηλέφωνο τοποθετήθηκε στην αντίθετη τσέπη από την κατεύθυνση της πτώσης. Πριν από τις πτώσεις οι εθελοντές υπέγραψαν ένα σχετικό έντυπο ανάληψης ευθύνης καθώς η παρουσία γιατρού δεν κατέστη δυνατή.

3.2.1 Εθελοντές

Όλοι οι εθελοντές που χρησιμοποιήθηκαν φορούσαν παντελόνι με εμπρός και πίσω τσέπες κατά τη διάρκεια των καταγραφών. Η θέση και ο προσανατολισμός του τηλεφώνου κατά τη διάρκεια των καταγραφών επιλέχθηκε από τον εκάστοτε εθελοντή σε μία προσπάθεια να γίνει όσο το δυνατό ρεαλιστικότερο το σύνολο δεδομένων που θα καταγραφούν.

Αναγνωριστικό εθελοντή(id)	Όνομα	Επώνυμο	Ηλικία	Ύψος (cm)	Βάρος (kgr)	Φύλο
1	Sub1	Sub1	32	180	85	Male
2	Sub2	Sub2	26	169	64	Male
3	Sub3	Sub3	26	164	55	Female
4	Sub4	Sub4	32	186	93	Male
5	Sub5	Sub5	36	160	50	Female
6	Sub6	Sub6	22	172	62	Female
7	Sub7	Sub7	25	189	80	Male
8	Sub8	Sub8	22	183	93	Male
9	Sub9	Sub9	30	177	102	Male
10	Sub10	Sub10	26	170	90	Female
11	Sub11	Sub11	26	168	80	Female
12	Sub12	Sub12	29	178	83	Male
13	Sub13	Sub13	24	177	62	Male
14	Sub14	Sub14	24	178	85	Male
15	Sub15	Sub15	25	173	82	Male
16	Sub16	Sub16	27	172	56	Female
17	Sub17	Sub17	25	173	67	Male
18	Sub18	Sub18	25	176	73	Male
19	Sub19	Sub19	25	161	63	Female
20	Sub20	Sub20	26	178	71	Male
21	Sub21	Sub21	25	180	70	Male

29	Sub29	Sub29	27	186	103	Male
30	Sub30	Sub30	47	172	90	Male
31	Sub31	Sub31	27	170	75	Male

Πίνακας 2: Εθελοντές

Οι εθελοντές που χρησιμοποιήθηκαν στις καταγραφές ήταν 17 άνδρες με ηλικία μεταξύ 22-47, βάρος 62-103kg και ύψος 1,69-1,89m και 7 γυναίκες με ηλικία 22-36, βάρος 50-90kg και ύψος 1,60-1,72m.

3.2.2 Πτώσεις

Τα χαρακτηριστικά κάθε τύπου πτώσης παρουσιάστηκαν σε κάθε εθελοντή και στη συνέχεια ο εθελοντής προσομοίωσε τον κάθε τύπο πτώσης τρεις φορές. Σε όλες τις πτώσεις υπήρξε παρόν είτε ο συγγραφέας είτε κάποιο μέλος του εργαστηρίου Βιοϊατρικής Πληροφορικής και Ηλεκτρονικής Υγείας του ΤΕΙ Κρήτης (Biomedical Informatics and eHealth Laboratory, BMI Lab, www.bmi.teicrete.gr).

Κωδικός	Τύπος πτώσης	Αριθμός δοκιμών	Διάρκεια	Περιγραφή
FOL	Forward-lying	3	10s	Πτώση μπροστά από όρθια θέση και χρησιμοποίηση των χεριών για απορρόφηση της πτώσης
FKL	Front-knees-lying	3	10s	Πτώση μπροστά από όρθια θέση- πρώτη κρούση στα γόνατα
SDL	Sideward-lying	3	10s	Πτώση στο πλάι από όρθια θέση λυγίζοντας τα πόδια
BSC	Back-sitting-chair	3	10s	Πτώση πίσω καθώς κάθεται σε καρέκλα

Πίνακας 3: Τύποι πτώσεων

3.2.2.1 Πτώση μπροστά με γόνατα

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην πίσω τσέπη και ενώ βρίσκεται επάνω στα στρώματα αφήνει το βάρος του ελεύθερο να πέσει προς τα εμπρός λυγίζοντας τα πόδια. Η πρώτη κρούση γίνεται με τα γόνατα και στη συνέχεια το σώμα του εθελοντή κατευθύνεται προς το στρώμα ώστε να είναι ξαπλωμένος και παραμένει εκεί μέχρι να τελειώσει η καταγραφή.

3.2.2.2 Πτώση μπροστά

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην πίσω τσέπη και ενώ βρίσκεται επάνω στα στρώματα αφήνει το βάρος του ελεύθερο να πέσει προς τα εμπρός λυγίζοντας τα πόδια, χωρίς όμως αυτά να ακουμπήσουν στο στρώμα. Η πρώτη κρούση γίνεται με τα χέρια τα οποία

βρίσκονται στο ύψος του στήθους. Ο εθελοντής παραμένει ξαπλωμένος στο στρώμα μετά την πτώση μέχρι να ολοκληρωθεί η καταγραφή.

3.2.2.3 Πτώση πίσω

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην μπροστά τσέπη και στέκεται με το στρώμα να βρίσκεται πίσω του. Ο εθελοντής πέφτει προς το στρώμα σαν να κάθεται σε μία φανταστική καρέκλα.. Η πρώτη κρούση γίνεται με την λεκάνη και στη συνέχεια με την πλάτη. Στην συνέχεια παραμένει εκεί μέχρι να τελειώσει η καταγραφή.

3.2.2.4 Πτώση στο πλάι

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην μπροστά τσέπη αλλά στην αντίθετη μεριά από αυτήν που θα πέσει. Στην συνέχεια αφήνει το σώμα του να πέσει προς το πλάι με πρώτη κρούση στα γόνατα.



Εικόνα 5: Πτώση στο πλάι

3.2.3 Καθημερινές δραστηριότητες (ADLs)

Οι καθημερινές δραστηριότητες έγιναν στο χώρο του ΤΕΙ Κρήτης. Η θέση και ο προσανατολισμός της συσκευής επάνω στον εθελοντή επιλέχθηκε από τον ίδιο τον εθελοντή κατά τη διάρκεια των καταγραφών με μόνο περιορισμό το κινητό τηλέφωνο να βρίσκεται στην μπροστινή τσέπη. Σε κάθε καταγραφή ο συγγραφέας περνούσε τις παραμέτρους στην εφαρμογή και στη συνέχεια το κινητό τηλέφωνο δινόταν στον εθελοντή και το τοποθετούσε στην τσέπη του.

Κωδικός	Τύπος δραστηριότητας	Αριθμός δοκιμών	Διάρκεια	Περιγραφή
STD	Standing	1	5m	Ορθοστασία κάνοντας μικρές κινήσεις
WAL	Walking	1	5m	Κανονικό περπάτημα
JOG	Jogging	3	30s	Τρέξιμο
JUM	Jumping	3	30s	Συνεχή άλματα
STU	Stairs up	6	10s	Ανέβασμα σκάλας (10 σκαλοπάτια)
STN	Stairs down	6	10s	Κατέβασμα σκάλας (10 σκαλοπάτια)
SCH	Sit chair	6	6s	Κάθισμα σε καρέκλα
CSI	Car-step in	6	6s	Είσοδος σε αυτοκίνητο
CSO	Car-step out	6	6s	Έξοδος από αυτοκίνητο

Πίνακας 4: Καθημερινές δραστηριότητες

3.2.3.1 Ορθοστασία (Standing)

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην μπροστά τσέπη του παντελονιού και στέκεται μέχρι να τελειώσει η καταγραφή. Επίσης σε αυτήν τη δραστηριότητα μπορεί να κάνει μικρές κινήσεις όπως να γυρίσει το σώμα του ενώ στέκεται ή να στηρίζει το βάρος του σε ένα από τα δύο πόδια.

3.2.3.2 Περπάτημα (Walking)

Μετοκινητό τηλέφωνο τοποθετημένο στην μπροστά τσέπη του εθελοντή ξεκινάει να περπατάει με τον δικό του ρυθμό. Ο ρυθμός των εθελοντών ήταν ούτε πολύ γρήγορος αλλά ούτε και πολύ αργός.

3.2.3.3 Τρέξιμο (Jogging)

Ο εθελοντής ξεκινάει να τρέχει μόλις ξεκινήσει η καταγραφή. Και πάλι ο ρυθμός του τρεξίματος δεν ήταν πολύ γρήγορος ή πολύ αργός. Το κινητό τοποθετήθηκε στην μπροστά τσέπη.

3.2.3.4 Άλματα (Jumping)

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην τσέπη του και ξεκινάει επιτόπια άλματα μέχρι να τελειώσει η καταγραφή.

3.2.3.5 Κάθισμα σε καρέκλα (Sitting to a chair)

Ο εθελοντής τοποθετεί το κινητό τηλέφωνο στην τσέπη του και στέκεται μπροστά σε μία καρέκλα. Μόλις ξεκινήσει η καταγραφή κάθεται στη καρέκλα και παραμένει καθιστός έως να τελειώσει η καταγραφή.

3.2.3.6 Είσοδος σε αυτοκίνητο (Step in a car)

Ο εθελοντής στέκεται έξω από το αυτοκίνητό με την πόρτα του αυτοκινήτου ανοιχτή. Όταν ξεκινάει η καταγραφή ο εθελοντής μπαίνει μέσα στο αυτοκίνητο και κάθεται στο κάθισμά. Όλες οι καταγραφές έγιναν στη θέση του συνοδηγού.

3.2.3.7 Έξοδος από αυτοκίνητο (Step out a car)

Ο εθελοντής κάθεται στη θέση του συνοδηγού. Μόλις ξεκινήσει η καταγραφή βγαίνει από το αυτοκίνητο και στέκεται δίπλα από αυτό μέχρι να τελειώσει η καταγραφή.

3.2.3.8 Ανέβασμα σκάλας (Ascending stairs)

Ο εθελοντής στέκεται μπροστά στα σκαλοπάτια. Η καταγραφή ξεκινάει και ο εθελοντής ξεκινάει να ανεβαίνει τα σκαλοπάτια. Όταν τελειώσουν τα σκαλοπάτια στέκεται μέχρι να τελειώσει η καταγραφή.

3.2.3.9 Κατέβασμα σκάλας (Descending Stairs)

Ο εθελοντής στέκεται μπροστά στα σκαλοπάτια. Η καταγραφή ξεκινάει και ο εθελοντής ξεκινάει να κατεβαίνει τα σκαλοπάτια. Όταν τελειώσουν τα σκαλοπάτια στέκεται μέχρι να τελειώσει η καταγραφή.

3.3 Αρχεία

3.3.1 Αρχικά δεδομένα

Τρία .txt αρχεία αποθηκεύονται για κάθε επανάληψη σε κάθε καταγραφή, ένα για το επιταχυσιόμετρο, ένα για το γυροσκόπιο και ένα για τον προσανατολισμό. Το κάθε αρχείο είναι δομημένο και περιλαμβάνει μία επικεφαλίδα και τα δεδομένα. Η επικεφαλίδα βρίσκεται στις πρώτες γραμμές του αρχείου και ολοκληρώνεται στην ετικέτα @DATA. Από αυτήν την ετικέτα και έπειτα ξεκινάνε τα δεδομένα που έχουν καταγράψει οι αισθητήρες. Στην επικεφαλίδα υπάρχουν πληροφορίες σχετικά με τους αισθητήρες, την ώρα που έγινε η καταγραφή όπως επίσης και πληροφορίες σχετικά με τον εθελοντή όπως φαίνεται και στην Εικόνα 6.

```
#Acceleration force along the x y z axes (including gravity).
#timestamp(ns),x,y,z(m/s^2)
#Datetime: 05/07/2013 11:33:54
#####
#Activity: 11 - FKL - Fall front knees lying - 10s
#Subject ID: 6
#First Name: sub6
#Last Name: sub6
#Age: 22
#Height (cm): 172
#Weight (kg): 62
#Gender: Female
#####

@DATA
2693088902000, 0.7182605, 9.825804, -1.1109096
2693289745000, 0.7565677, 9.720459, -1.0247183
2693489138000, 0.82360536, 9.835381, -0.8714894
2693501929000, 0.79487497, 9.80665, -0.9864111
2693512780000, 0.79487497, 9.797073, -1.1013328
2693524453000, 0.7757214, 9.758766, -1.0534488
2693534002000, 0.7757214, 9.768343, -1.0821792
2693547322000, 0.7565677, 9.730036, -1.0630256
2693558625000, 0.78529817, 9.710882, -1.0342951
2693573179000, 0.78529817, 9.739613, -0.93852705
2693584137000, 0.8331822, 9.80665, -0.90021986
2693593722000, 0.842759, 9.854534, -0.8714894
2693603934000, 0.8331822, 9.864111, -0.79487497
2693615990000, 0.79487497, 9.873688, -0.7757214
2693625887000, 0.7374141, 9.854534, -0.7565677
2693638532000, 0.6895301, 9.854534, -0.7661445
2693649208000, 0.63206923, 9.835381, -0.7469909
2693662552000, 0.63206923, 9.797073, -0.7469909
```

Εικόνα 6: Αρχείο με τα δεδομένα του επιταχυνσιόμετρου

Κάθε αρχείο στην περιοχή των δεδομένων είναι χωρισμένο σε τέσσερις στήλες. Κάθε στήλη χωρίζεται από την άλλη με κόμμα «,». Η πρώτη στήλη, η οποία είναι κοινή σε όλα τα αρχεία, είναι το timestamp. Το timestamp δηλώνει την ακριβή ώρα που πάρθηκαν οι μετρήσεις από τον αισθητήρα και η τιμή του είναι σε nanoseconds. Οι τρεις επόμενες στήλες διαφέρουν σε κάθε αρχείο ανάλογα με τον τύπο του αισθητήρα.

- Accelerometer: timestamp (ns), x, y, z (m/s^2)
- Gyroscope: timestamp(ns), x, y, z (rad/s)
- Orientation: timestamp(ns), azimuth, pitch, roll (degrees)

Πιο συγκεκριμένα ο αισθητήρας επιτάχυνσης μετράει την επιτάχυνση της συσκευής στους τρεις άξονες : z, y, z σε m/sec^2 . Το gyroscope μετράει την γωνιακή ταχύτητα (angular velocity) της συσκευής σε rad/sec και το orientation μετράει την γωνία σε μοίρες γύρο από τους τρεις άξονες της συσκευής (azimuth, pitch, roll).

3.3.2 Ονοματολογία αρχείων

Τα ονόματα των αρχείων που δημιουργήθηκαν από την εφαρμογή κατά τη διάρκεια των καταγραφών έχουν συγκεκριμένη μορφή η οποία περιγράφεται στη συνέχεια. Κάθε αρχείο .txt ξεκινάει με τους τρεις χαρακτήρες του κωδικού της δραστηριότητας ή της πτώσης (Πίνακας 3& Πίνακας 4). Έπειτα ακολουθούν οι χαρακτήρες του κωδικού του αισθητήρα που προσδιορίζουν ποίος αισθητήρας αντιστοιχεί σε κάθε αρχείο (Πίνακας 5). Στη συνέχεια το αναγνωριστικό του εθελοντή (Πίνακας 2) και τέλος ο αριθμός της δοκιμής. Όλες οι παραπάνω πληροφορίες χωρίζονται με τον χαρακτήρα της κάτω-παύλας (underscore) «_». Για παράδειγμα, όταν ο εθελοντής με αναγνωριστικό 5 πραγματοποίησε την δεύτερη δοκιμή της πίσω-πτώσης ενώ κάθεται σε καρέκλα η εφαρμογή δημιούργησε τα παρακάτω τρία αρχεία:

1. BSC_acc_5_2.txt
2. BSC_gyro_5_2.txt
3. BSC_ori_5_2.txt

Γενικότερα τα ονόματα αρχείων έχουν την εξής μορφοποίηση:

- [ActivityCode]_[SensorCode]_[SubjectsID]_[TrialID].txt

Κωδικός αισθητήρα	Τύπος αισθητήρα	Τιμές	Περιγραφή
acc	Accelerometer	timestamp(ns),x, y, z(m/s ²)	Acceleration force along the x y z axes (including gravity)
gyro	Gyroscope	timestamp(ns),x, y, z(rad/s)	Rate of rotation around the x y z axes (Angular velocity)
ori	Orientation	timestamp(ns),azimuth, pitch, roll(degrees)	Angle around the z x y axes

Πίνακας 5: Αισθητήρες

4 Μεθοδολογία Υλοποίησης και Ανάπτυξη Εφαρμογών

4.1 Πλατφόρμες και εργαλεία που χρησιμοποιήθηκαν

4.1.1 Eclipse

Το Eclipse¹ είναι ένα περιβάλλον ανάπτυξης λογισμικού το οποίο είναι γραμμένο σε γλώσσα Java. Είναι ένα από τα πιο δημοφιλή IDE για την ανάπτυξη εφαρμογών σε Java αλλά και για άλλες γλώσσες προγραμματισμού με την χρήση των plugins που μπορούμε να του εγκαταστήσουμε. Για την ανάπτυξη εφαρμογών για συσκευές με λειτουργικό σύστημα Android χρησιμοποιούμε το ADT plugin το οποίο συνεργάζεται με το Eclipse. Η έκδοση που χρησιμοποιήθηκε είναι η 3.7.2 με την ονομασία Eclipse Indigo.

4.1.2 Android

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα (open source) για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την OpenHandsetAlliance². Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Οι εφαρμογές γράφονται σε γλώσσα Java χρησιμοποιώντας βιβλιοθήκες του Android.

Λειτουργίες Οθόνης	Η πλατφόρμα είναι προσαρμόσιμη σε μεγαλύτερη ανάλυση (VGA), διδιάστατες ψηφιακές γραφικές βιβλιοθήκες, τρισδιάστατα γραφικά βασισμένα στην OpenGL ES 1.0 έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης "έξυπνων" συσκευών κινητής τηλεφωνίας.
Αποθήκευση Δεδομένων	Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης
Συνδεσιμότητα	Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, και Wi-Fi.

¹<https://www.eclipse.org/>

²<http://www.openhandsetalliance.com/>

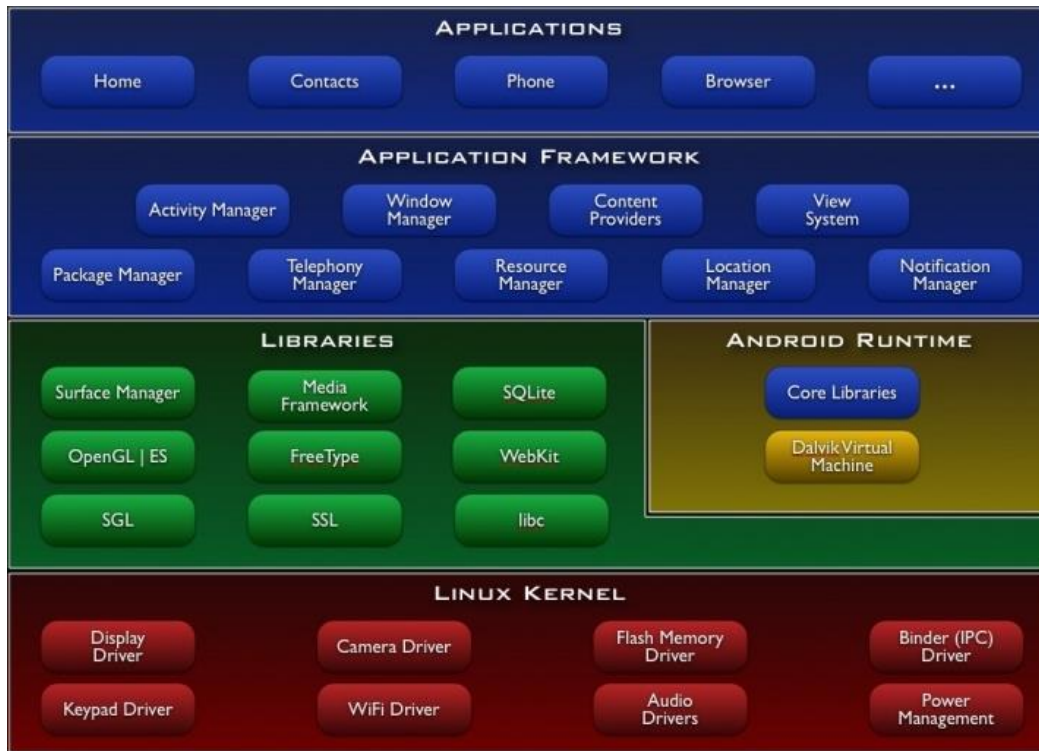
Αποστολή μηνυμάτων	SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.
Περιήγηση στον Ιστό	Για την περιήγηση στον ιστό το Android διαθέτει φυλλομετρητή (browser) βασισμένο στην ανοιχτή τεχνολογία WebKit.
Υποστήριξη Java	Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik, η οποία αποτελεί εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένης για χρήση σε φορητές συσκευές, παρόλο που δεν είναι πρότυπη εικονική μηχανή Java.
Υποστήριξη Πολυμέσων	Το λειτουργικό Android υποστηρίζει τις ακόλουθα μορφές ήχου, στατικής και κινούμενης εικόνας: H.263, H.264 (σε 3GP ή MP4 container), MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.
Επιπλέον υποστήριξη υλικού	Το λειτουργικό Android μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.

Πίνακας 6: Χαρακτηριστικά και λειτουργίες του Android

Τα πλούσια χαρακτηριστικά του Android (δες Πίνακας 6) οδήγησε στο να γίνει το Android η πλατφόρμα που χρησιμοποιείται περισσότερο σε έξυπνα κινητά και ταμπλέτες (tablets) σε όλο τον κόσμο ξεπερνώντας το ανταγωνιστικό Symbian. Το Android έχει το 75% της αγοράς έξυπνων κινητών στο τρίτο τέταρτο του 2012 με 750 εκατομμύρια συσκευές ενεργοποιημένες και 1,5 εκατομμύριο συσκευές να ενεργοποιούνται κάθε μέρα [23].

4.1.2.1 Αρχιτεκτονική

Το διάγραμμα που ακολουθεί δείχνει τα στοιχεία που αποτελείται το λειτουργικό σύστημα Android. Κάθε επίπεδο περιγράφεται αναλυτικά παρακάτω.



Εικόνα 7: Αρχιτεκτονική Android

4.1.2.1.1 Εφαρμογές (Applications)

Το Android έχει ένα σετ από εφαρμογές οι οποίες μπορεί να είναι πελάτης ηλεκτρονικού ταχυδρομείου (email client), πρόγραμμα επεξεργασίας SMS, ημερολόγιο, χάρτες, πρόγραμμα πλοήγησης (browser), επεξεργαστής επαφών τηλεφώνου, κ.α. Όλες οι εφαρμογές είναι γραμμένες σε γλώσσα προγραμματισμού Java.

4.1.2.1.2 Πλαίσιο εφαρμογών (Application Framework)

Παρέχοντας μία ανοιχτή πλατφόρμα ανάπτυξης, το Android προσφέρει την δυνατότητα στους προγραμματιστές να υλοποιήσουν ένα πλήθος από καινοτόμες εφαρμογές. Τους δίνεται η δυνατότητα να χρησιμοποιήσουν το υλικό (hardware) της συσκευής, να πάρουν πληροφορίες θέσης, να τρέξουν υπηρεσίες στο παρασκήνιο (background services), να ενεργοποιήσουν ειδοποιήσεις, να τις προσθέσουν στη περιοχή ειδοποιήσεων και πολλά άλλα.

Οι προγραμματιστές μπορούν να έχουν πρόσβαση στο framework API που χρησιμοποιείται από τις εφαρμογές. Η αρχιτεκτονική της κάθε εφαρμογής είναι σχεδιασμένη έτσι ώστε να μπορούν να επαναχρησιμοποιούνται κομμάτια από διάφορες εφαρμογές.

Όλες οι εφαρμογές είναι σετ από υπηρεσίες και συστήματα:

- Ένα πλήθος από όψεις (views) που χρησιμοποιούνται για το χτίσιμο μίας εφαρμογής. Για παράδειγμα λίστες, πλέγματα, πεδία κειμένου, κουμπιά, ή ακόμα και έναν ενσωματωμένο πρόγραμμα πλοήγησης
- Παροχές περιεχομένου οι οποίοι ενεργοποιούν τις εφαρμογές να προσπελάσουν δεδομένα από άλλες εφαρμογές όπως οι επαφές τηλεφώνου.

- Ένας διαχειριστής αρχείων πηγών που προσφέρει πρόσβαση σε αρχεία όπως μεταφρασμένα strings, γραφικά και διατάξεις (layouts).
- Έναν διαχειριστή ειδοποιήσεων (notification manager) που επιτρέπει στις εφαρμογές να εμφανίζουν παραμετροποιημένες ειδοποιήσεις στη μπάρα κατάστασης.
- Έναν διαχειριστή δραστηριοτήτων (activity manager) ο οποίος διαχειρίζεται το κύκλο ζωής των activities και δίνει τη δυνατότητα περιήγησης στις εφαρμογές.

4.1.2.1.3 Βιβλιοθήκες (Libraries)

Το Android περιέχει ένα σετ από C/C++ βιβλιοθήκες με διάφορα συστατικά του συστήματος Android. Αυτά τα συστατικά δίνονται στους προγραμματιστές μέσω του πλαισίου εφαρμογής (application framework). Μερικά από αυτά είναι:

- **System C library:** Βιβλιοθήκη της C ρυθμισμένη για ενσωματωμένες συσκευές linux.
- **Media libraries:** είναι οι βιβλιοθήκες που υποστηρίζουν πολλά πολυμεσικά αρχεία όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG
- **Surface manager:** Διαχειρίζεται την πρόσβαση στα γραφικά 2D και 3D.
- **Libweb core:** Μοντέρνα μηχανή πλοήγησης web που περιλαμβάνει και το πρόγραμμα πλοήγησης του Android και το ενσωματωμένο webview.
- **Sgl:** γραφικά 2D
- **3d libraries:** γραφικά 3D openGL.
- **Free type:** γραμματοσειρές bitmap και vectors
- **Sql lite:** Μία ισχυρή και ελαφριά σχεσιακή βάση δεδομένων διαθέσιμη για όλες τις εφαρμογές.

4.1.2.1.4 Android Runtime

Το Android περιέχει ένα σετ βιβλιοθηκών που προσφέρουν τις περισσότερες λειτουργίες που είναι διαθέσιμες στην γλώσσα προγραμματισμού Java.

Κάθε εφαρμογή Android τρέχει την δική της διεργασία με το δικό της στιγμιότυπο στη εικονική μηχανή Dalvik. Η DVM είναι γραμμένη έτσι ώστε μία συσκευή να τρέχει πολλές VMs αποτελεσματικά. Η DVM τρέχει αρχεία dalvik εκτελέσιμα (.dex) τύπου τα οποία είναι σχεδιασμένα για μικρή μνήμη. Η DVM στηρίζεται στον πυρήνα του linux για λειτουργικότητες όπως το threading και το χαμηλού επιπέδου διαχείριση μνήμης.

4.1.2.1.5 Πυρήνας Linux (Linux Kernel)

Το Android στηρίζεται στην έκδοση Linux 2.6 για υπηρεσίες συστήματος όπως ασφάλεια, διαχείριση μνήμης, διαχείριση διεργασιών, στοίβα δικτύου, και μοντέλα οδηγών. Ο πυρήνας Linux λειτουργεί επίσης ως ενδιάμεσο επίπεδο ανάμεσα σε υλικό και λογισμικό. [24]

4.1.3 Matlab

Το Matlab είναι ένα αλληλεπιδραστικό περιβάλλον ανάπτυξης και μία γλώσσα υψηλού επιπέδου για αριθμητικούς υπολογισμούς, οπτικοποιήσεις και προγραμματισμό. Χρησιμοποιώντας το Matlab, μπορούμε να αναλύσουμε δεδομένα, να αναπτύξουμε αλγόριθμους και να κατασκευάσουμε μοντέλα και εφαρμογές. Τα εργαλεία της γλώσσας και οι ενσωματωμένες μαθηματικές συναρτήσεις βοηθούν στο να ερευνηθούν πολλαπλές

προσεγγίσεις και να φτάσουμε γρηγορότερα σε μία λύση απ' ό,τι με τα λογιστικά φύλλα και τις παραδοσιακές γλώσσες προγραμματισμού, όπως οι C/C++ και η Java.

Το Matlab μπορεί να χρησιμοποιηθεί για ένα πλήθος εφαρμογών, περιλαμβάνοντας την επεξεργασία σήματος και επικοινωνίες, επεξεργασία εικόνας και βίντεο, συστήματα ελέγχου, έλεγχος και μέτρηση υπολογιστικού κόστους και υπολογιστική βιολογία. Περισσότεροι από ένα εκατομμύριο μηχανικοί και επιστήμονες στην βιομηχανία αλλά και στην εκπαίδευση χρησιμοποιούν το Matlab. [25]

4.1.4 WEKA

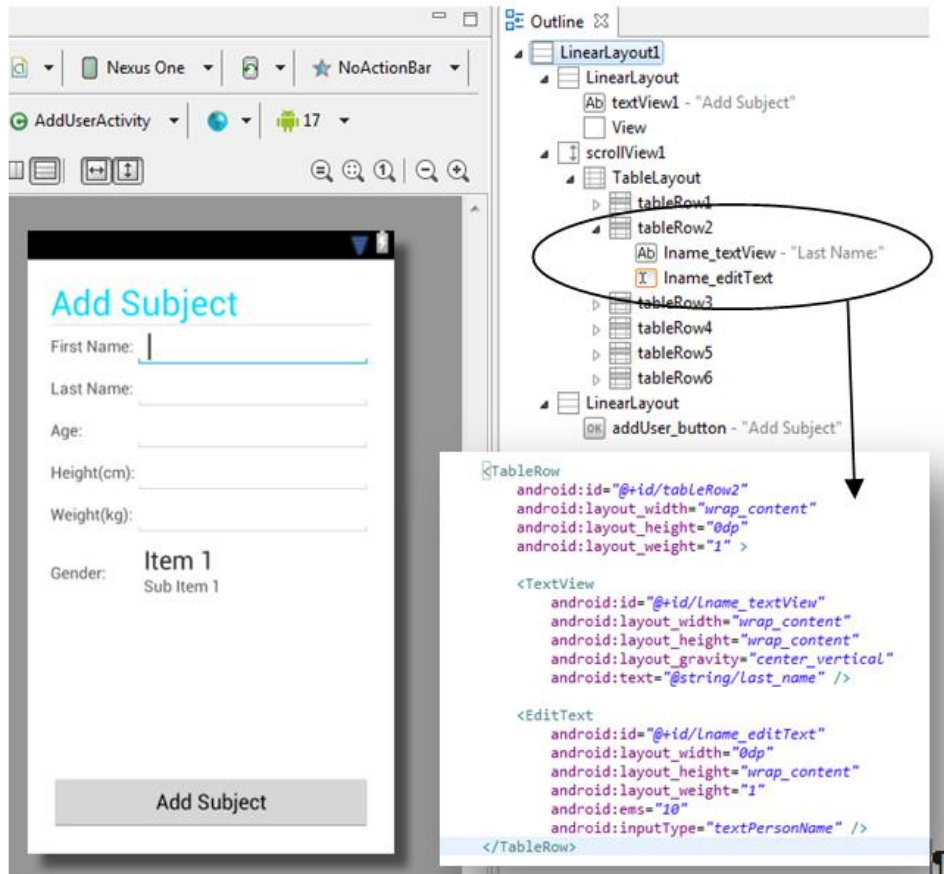
Το Waikato Environment for Knowledge Analysis (WEKA) είναι ένα λογισμικό ανοιχτού κώδικα γραμμένο σε Java το οποίο ξεκίνησε το 1992. Είναι αναγνωρισμένο και ευρέως γνωστό και διαδεδομένο στους πανεπιστημιακούς αλλά και στους επιχειρηματικούς κύκλους. Το WEKA χρησιμοποιείται ευρέως σε έρευνες σχετικές με την επεξεργασία και συλλογή δεδομένων. [26]

4.2 Ανάπτυξη εφαρμογής Android

4.2.1 Δημιουργία γραφικής διεπαφής

Για την δημιουργία της γραφικής διεπαφής της εφαρμογής χρησιμοποιήθηκε το Eclipse. Κάθε διαφορετική οθόνη είναι και ένα διαφορετικό Activity [27] όπως και μία διαφορετική διάταξη (layout). Το κάθε layout είναι ένα αρχείο xml το οποίο μέσα περιέχει τα διάφορα συστατικά που θα εμφανίζονται στην οθόνη. Συνολικά δημιουργήθηκαν 11 διατάξεις (layouts) με διάφορα γραφικά στοιχεία μέσα το καθένα.

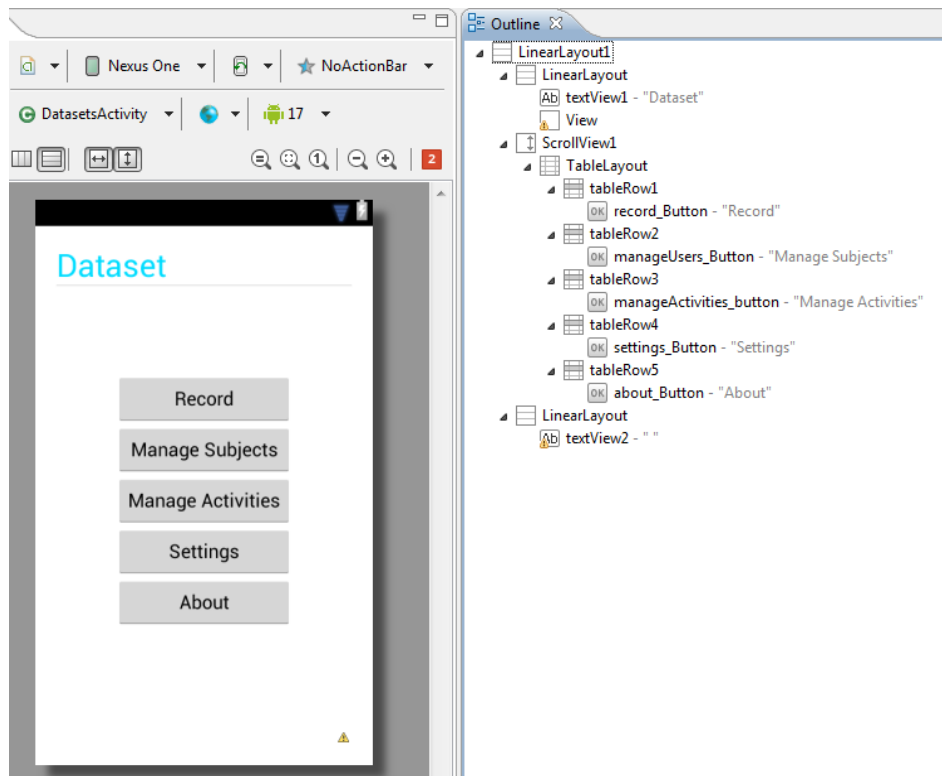
Στην Εικόνα 8 φαίνεται το στιγμιότυπο από τον σχεδιασμό της οθόνης της εφαρμογής όταν προσθέτουμε κάποιον χρήστη. Βλέπουμε ότι η φόρμα περιέχει τα χαρακτηριστικά του εθελοντή με τα πεδία Age, Height και Weight. Τα πεδία αυτά μπορούν να δεχτούν μόνο αριθμούς με μέγιστο τρεις χαρακτήρες. Το πεδίο Gender είναι τύπου αναδιπλούμενης λίστας (drop-down list) και μπορούμε να επιλέξουμε μόνο τις επιλογές male ή female.



Εικόνα 8: Γραφική υλοποίηση προσθήκης χρήστη

Στην Εικόνα 9 φαίνεται η γραφική διεπαφή της αρχικής οθόνης. Επιλέχθηκε ως βασική διάταξη (Root Layout) αυτή της μορφής `LinearLayout`, το οποίο είναι μία διάταξη που οργανώνει τα παιδιά της σε μία στήλη ή μία γραμμή. Η κατεύθυνση της γραμμής μπορεί να ρυθμιστεί με την κλήση `setOrientation()`. Ο προεπιλεγμένος προσανατολισμός είναι οριζόντιος. Μέσα σε αυτό υπάρχουν 3 ακόμα στοιχεία. Επάνω βρίσκεται η γραμμική διάταξη (`LinearLayout`) που μέσα σε αυτή υπάρχει ο τίτλος της εφαρμογής και η οριζόντια γραμμή. Κάτω από αυτό υπάρχει η κυλιόμενη όψη (`ScrollView`) το οποίο περιέχει μία διάταξη πίνακα (`TableLayout`) μέσα στο οποίο υπάρχουν σε σειρές τα κουμπιά της εφαρμογής. Χρησιμοποιήθηκε η κυλιόμενη όψη (`ScrollView`) έτσι ώστε εάν η εφαρμογή τρέξει σε μικρότερη οθόνη να μπορεί ο χρήστης να περιηγηθεί σε όλα τα κουμπιά και να τα επιλέξει. Τέλος στο κάτω μέρος υπάρχει άλλη μία γραμμική διάταξη (`LinearLayout`) με μία όψη κειμένου (`textView`) μέσα στην οποία μπορούμε να τυπώσουμε κάποιο μήνυμα στον χρήστη.

Με τον ίδιο τρόπο που περιγράφηκε προηγουμένως και ακολουθώντας την δομή αυτήν, δηλαδή επικεφαλίδα, κυρίως μέρος και κάτω μέρος φτιάχτηκαν και τα υπόλοιπα (συνολικά 11) διατάξεις (`layouts`) της εφαρμογής. Οι διατάξεις (`layouts`) της εφαρμογής δεν έχουν καμία λειτουργία, απλά καθορίζουν πώς εμφανίζονται τα διάφορα συστατικά της εφαρμογής στην οθόνη.



Εικόνα 9: Δημιουργία της διάταξης (layout) για την αρχική οθόνη

4.2.2 Συλλογή δεδομένων από αισθητήρες

Η συλλογή δεδομένων από τους αισθητήρες του smartphone είναι η βασικότερη λειτουργία της εφαρμογής. Για να πάρουμε δεδομένα από τους αισθητήρες θα πρέπει να χρησιμοποιήσουμε την κλάση `SensorManager` του Android και να καταχωρήσουμε (register) τους αισθητήρες που θα χρησιμοποιήσουμε. Στην παρακάτω εικόνα φαίνονται οι μέθοδοι `registerSensors()` και `unregisterSensors()` οι οποίες χρησιμοποιούνται για να ενεργοποιήσουν τους αισθητήρες και να τους απενεργοποιήσουν. Η απενεργοποίηση των αισθητήρων όταν η εφαρμογή σταματήσει να δουλεύει είναι πολύ σημαντική διότι εάν δεν απενεργοποιηθούν η εφαρμογή θα συνεχίσει να τους χρησιμοποιεί καταναλώνοντας υπολογιστικούς πόρους. Παρατηρούμε στη μέθοδο `registerSensors()` ότι καλείται η μέθοδος `registerListener()` με ορίσματα α) τον listener β) τον αισθητήρα και γ) τον ρυθμό δειγματοληψίας του αισθητήρα. Ο ρυθμός δειγματοληψίας που επιλέχθηκε είχε την τιμή `SensorManager.SENSOR_DELAY_FASTEST` και στους τρεις αισθητήρες καθώς αυτή είναι ο μέγιστος ρυθμός δειγματοληψίας που προσφέρει το Android. Οι δύο αυτές μέθοδοι καλούνται πριν και μετά από κάθε καταγραφή.

```

private void registerSensors() {
    if (sensorManager != null) {
        sensorManager.registerListener(this, accelerometer,
            SensorManager.SENSOR_DELAY_FASTEST);
        sensorManager.registerListener(this, orientation,
            SensorManager.SENSOR_DELAY_FASTEST);
        sensorManager.registerListener(this, gyroscope,
            SensorManager.SENSOR_DELAY_FASTEST);
    }
}

private void unregisterSensors() {
    if (sensorManager != null) {
        sensorManager.unregisterListener(this);
    }
}

```

Εικόνα 10: Μέθοδοι ενεργοποίησης και απενεργοποίησης αισθητήρων

4.2.3 Διαχείριση Εθελοντών και Δραστηριοτήτων

Άλλη μία βασική λειτουργία της εφαρμογής είναι η διαχείριση των εθελοντών και των δραστηριοτήτων. Για την λειτουργία αυτήν δημιουργήθηκε μία τοπική βάση δεδομένων στην εφαρμογή μέσω της οποίας μπορούμε να προσθέτουμε να διαγράφουμε και να επεξεργαζόμαστε χρήστες και δραστηριότητες. Η σημασία της υλοποίησης αυτής της λειτουργίας είναι ότι κατά τη διάρκεια των καταγραφών είναι δύσκολο να περνάμε συνεχώς τα δεδομένα του χρήστη και επιπλέον να μπορούμε εύκολα από την εφαρμογή να πραγματοποιήσουμε αλλαγές τόσο στους εθελοντές αλλά και στις καθημερινές δραστηριότητες και πτώσεις.

Παρακάτω περιγράφονται οι κλάσεις User και HumanActivity. Οι κλάσεις αυτές φαίνονται και στην Εικόνα 11: και Εικόνα 12.

Η κλάση User περιγράφει τον χρήστη ο οποίος θα εκτελέσει κάποια δραστηριότητα. Οι *Abbate et al.* [2] περιγράφουν ότι τα χαρακτηριστικά των εθελοντών θα πρέπει να είναι φύλο, ηλικία, βάρος, ύψος και ο δείκτης μάζας σώματος (Body Mass Index, BMI). Ο δείκτης μάζας σώματος υπολογίζεται από το ύψος και το βάρος. Επιπλέον χαρακτηριστικά που προστέθηκαν στην εφαρμογή είναι το όνομα και το επώνυμο του εθελοντή, καθώς και το αναγνωριστικό (id) έτσι ώστε κάθε εθελοντής να έχει το μοναδικό αναγνωριστικό για την αλληλεπίδραση της εφαρμογής με την βάση δεδομένων.

```

public class User {
    private int id;
    private String fname;
    private String lname;
    private String gender;
    private int age;
    private int height;
    private int weight;

    public User(int id, String fname, String lname, String gender, int age, int height, int weight) {
        this.id = id;
        this.fname = fname;
        this.lname = lname;
        this.gender = gender;
        this.age = age;
        this.height = height;
        this.weight = weight;
    }

    public User(String fname, String lname, String gender, int age, int height, int weight) {
        this.fname = fname;
        this.lname = lname;
        this.gender = gender;
        this.age = age;
        this.height = height;
        this.weight = weight;
    }
}

```

Εικόνα 11: Δημιουργία κλάσης User

Η κλάση HumanActivity περιγράφει μία δραστηριότητα που θα καταγράψει ο εθελοντής. Τα πεδία της είναι το αναγνωριστικό (id) το οποίο είναι μοναδικό για κάθε δραστηριότητα. Το σύμβολο ο τίτλος και η περιγραφή της δραστηριότητας τα οποία περιγράφουν ακριβώς τη δραστηριότητα. Ο τύπος της δραστηριότητας, που μπορεί να είναι είτε ADL είτε Fall ανάλογα με το activity. Και τέλος τα δευτερόλεπτα που διαρκεί η δραστηριότητα.

```

public class HumanActivity {
    private int id;
    private String symbol;
    private String title;
    private String Description;
    private String type;
    private int seconds;

    public HumanActivity(int id, String symbol, String title, String Description, String type, int seconds) {
        this.id = id;
        this.symbol = symbol;
        this.title = title;
        this.Description = Description;
        this.type = type;
        this.seconds = seconds;
    }

    public HumanActivity(String symbol, String title, String Description, String type, int seconds) {
        this.symbol = symbol;
        this.title = title;
        this.Description = Description;
        this.type = type;
        this.seconds = seconds;
    }
}

```

Εικόνα 12: Δημιουργία της κλάσης HumanActivity

4.2.3.1 Δημιουργία Βάσης δεδομένων

Η κλάση που δημιουργήθηκε είναι η DatabaseHandler η οποία είναι υποκλάση της SQLiteOpenHelper. Στην Εικόνα 13 φαίνεται ο κώδικας ο οποίος περιγράφει την κλάση αυτήν. Μέσα στην DatabaseHandler δημιουργήθηκαν οι μέθοδοι οι οποίοι διαχειρίζονται

τους εθελοντές και της δραστηριότητες καθώς και οι μέθοδοι (κάποιες από αυτές της SQLiteOpenHelper) οι οποίες δημιουργούσαν την βάση δεδομένων κατά την πρώτη εκκίνηση της εφαρμογής. Ο πίνακας user της βάσης δεδομένων αντιστοιχεί στους εθελοντές ενώ ο πίνακας activity στις δραστηριότητες και πτώσεις. Οι πτώσεις διαφέρουν από τις δραστηριότητες στο πεδίο type το οποίο παίρνει την τιμή ADL για τις καθημερινές δραστηριότητες και Fall για τις πτώσεις.

```
public class DatabaseHandler extends SQLiteOpenHelper {

    Context context;
    // Database Version
    private static final int DATABASE_VERSION = 1;

    // Database Name
    private static final String DATABASE_NAME = "datasetsDB";

    // tables names
    private static final String TABLE_USER = "user";
    private static final String TABLE_ACTIVITY = "activity";

    // user Table Columns names
    private static final String USER_ID = "id";
    private static final String USER_FNAME = "fname";
    private static final String USER_LNAME = "lname";
    private static final String USER_GENDER = "gender";
    private static final String USER_AGE = "age";
    private static final String USER_HEIGHT = "height";
    private static final String USER_WEIGHT = "weight";

    // activities Table Columns names
    private static final String ACT_ID = "id";
    private static final String ACT_SYMBOL = "symbol";
    private static final String ACT_TITLE = "title";
    private static final String ACT_DESC = "desc";
    private static final String ACT_SEC = "sec";
    private static final String ACT_TYPE = "type";

    public DatabaseHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.context = context;
    }
}
```

Εικόνα 13: Δημιουργία κλάσης DatabaseHandler

4.2.3.2 Αρχικοποίηση δεδομένων

Κατά την πρώτη εκκίνηση της εφαρμογής δημιουργήθηκαν 2 αρχεία JSON από τα οποία η DatabaseHandler έπαιρνε τους εθελοντές και τις δραστηριότητες που υπήρχαν σε αυτά. Αυτό έγινε για να περαστούν εύκολα οι δραστηριότητες και οι εθελοντές στην εφαρμογή και για να υπάρχει η ευκολία σε πιθανές νέες καταγραφές στο μέλλον να μπορεί αυτή η λίστα των δραστηριοτήτων ή των εθελοντών να αλλάξει εύκολα και να μην χρειάζεται να περνιούνται ξανά τα δεδομένα μέσα από την εφαρμογή.

```

activities.json
1 {
2   "activities": [
3     {
4       "id": 1,
5       "symbol": "STD",
6       "title": "Standing",
7       "description": "Standing with subtle movements",
8       "type": "adl",
9       "seconds": 300
10    },
11    {
12      "id": 2,
13      "symbol": "WAL",
14      "title": "Walking",
15      "description": "Normal walking",
16      "type": "adl",
17      "seconds": 300
18    },
19    {
20      "id": 3,
21      "symbol": "JOG",
22      "title": "Jogging",
23      "description": "Jogging",
24      "type": "adl",
25      "seconds": 30
26    },
27    {
28      "id": 4,
29      "symbol": "JUM",
30      "title": "Jumping",
31      "description": "Continuous jumping",
32      "type": "adl",
33      "seconds": 30
34    }
35  ]
36 }

subjects.json
1 {
2   "subjects": [
3     {
4       "id": 1,
5       "fName": "sub1",
6       "lName": "sub1",
7       "age": 32,
8       "height": "180",
9       "weight": 85,
10      "gender": "Male"
11    },
12    {
13      "id": 2,
14      "fName": "sub2",
15      "lName": "sub2",
16      "age": 26,
17      "height": "169",
18      "weight": 64,
19      "gender": "Male"
20    },
21    {
22      "id": 3,
23      "fName": "sub3",
24      "lName": "sub3",
25      "age": 26,
26      "height": "164",
27      "weight": 55,
28      "gender": "Female"
29    },
30    {
31      "id": 4,
32      "fName": "sub4",
33      "lName": "sub4",
34      "age": 32,
35    }
36  ]
37 }

```

Εικόνα 14: Αρχεία JSON με τους εθελοντές και της δραστηριότητες

4.2.4 Εγγραφή δεδομένων σε αρχεία

Η κλάση WriteToFile (Εικόνα 15) χρησιμοποιείται για να δημιουργήσουμε ένα αρχείο και να γράψουμε δεδομένα σε αυτό. Παρακάτω φαίνεται ο κατασκευαστής της κλάσης WriteToFile. Όταν δημιουργήσουμε ένα στιγμιότυπο της κλάσης αυτής με την εντολή:

```
WriteToFile wtfAcc = new WriteToFile(filename, subfolder);
```

```

public WriteToFile(String filename, String folder) {
    this.filename = filename;

    File root = Environment.getExternalStorageDirectory();

    if (root.canWrite()) {
        System.out.println("ok to write");
        File fdDirectory = new File(root + "/Dataset/" + folder + "/");
        fdDirectory.mkdirs();
        File outFile = new File(fdDirectory, this.filename);

        try {
            FileOutputStream fos = new FileOutputStream(outFile);
            out = new BufferedWriter(new OutputStreamWriter(fos));
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    } else {
        System.out.println("i cannot write");
    }
}

```

Εικόνα 15: Υλοποίηση κλάσης WriteToFile

Δημιουργείται ένα αρχείο όπου μπορούμε να γράψουμε μέσα σε αυτό. Η παραπάνω εντολή δημιουργεί ένα στιγμιότυπο της κλάσης WriteToFile με όνομα wtfAcc. Ο κατασκευαστής της κλάσης παίρνει ορίσματα δύο μεταβλητές String οι οποίες στην περίπτωση μας είναι το «filename» και το «subfolder». Η μεταβλητή filename είναι το όνομα του αρχείου που θα δημιουργηθεί και η μεταβλητή subfolder είναι ο υποφάκελος που θα δημιουργηθεί εάν δεν υπάρχει μέσα στο φάκελο Dataset της εξωτερικής μνήμης του κινητού τηλεφώνου.

Κατά τη διάρκεια της καταγραφής σε κάθε event που έρχεται από τους αισθητήρες εκτελείται η onSensorChanged. Εδώ είναι το σημείο όπου γράφονται τα δεδομένα των αισθητήρων στα ξεχωριστά αρχεία καλώντας κάθε φορά την μέθοδο write() από την κλάση WriteToFile (Εικόνα 16).

```
@Override
public void onSensorChanged(SensorEvent event) {
    switch (event.sensor.getType()) {
        case Sensor.TYPE_ACCELEROMETER:
            // timestamp, x, y, z
            wtfAcc.write(event.timestamp + ", " + event.values[0] + ", "
                + event.values[1] + ", " + event.values[2] + "\n");
            break;
        case Sensor.TYPE_ORIENTATION:
            // timestamp, azimuth, pitch, roll
            wtfOr.write(event.timestamp + ", " + event.values[0] + ", "
                + event.values[1] + ", " + event.values[2] + "\n");
            break;
        case Sensor.TYPE_GYROSCOPE:
            // timestamp, x, y, z
            wtfGyro.write(event.timestamp + ", " + event.values[0] + ", "
                + event.values[1] + ", " + event.values[2] + "\n");
            break;
        default:
            break;
    }
}
```

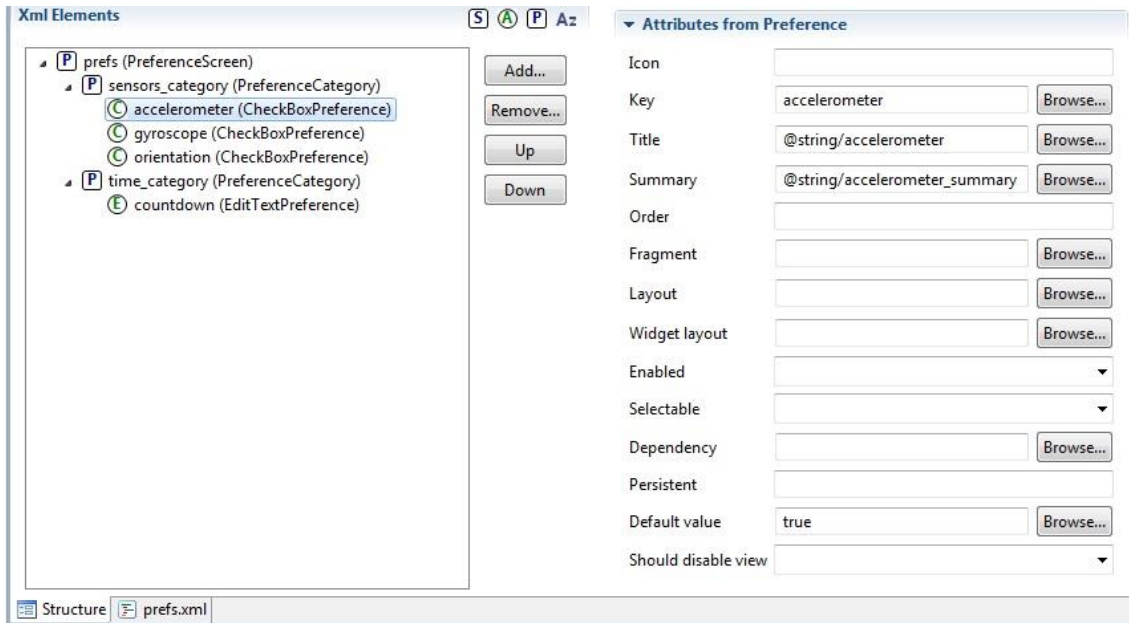
Εικόνα 16: Υλοποίηση της μεθόδου onSensorChanged()

4.2.5 Ρυθμίσεις Συστήματος

Η εφαρμογή καταγραφής δίνει την δυνατότητα στον χρήστη να αλλάξει παραμέτρους, όπως να επιλέξει ποιοι αισθητήρες θα χρησιμοποιηθούν στην καταγραφή και πόσος θα είναι ο χρόνος που η εφαρμογή θα περιμένει μέχρι να ξεκινήσει η καταγραφή. Για να υλοποιηθούν οι ρυθμίσεις φτιάχτηκε ένα αρχείο xml με το γραφικό εργαλείο του eclipse.

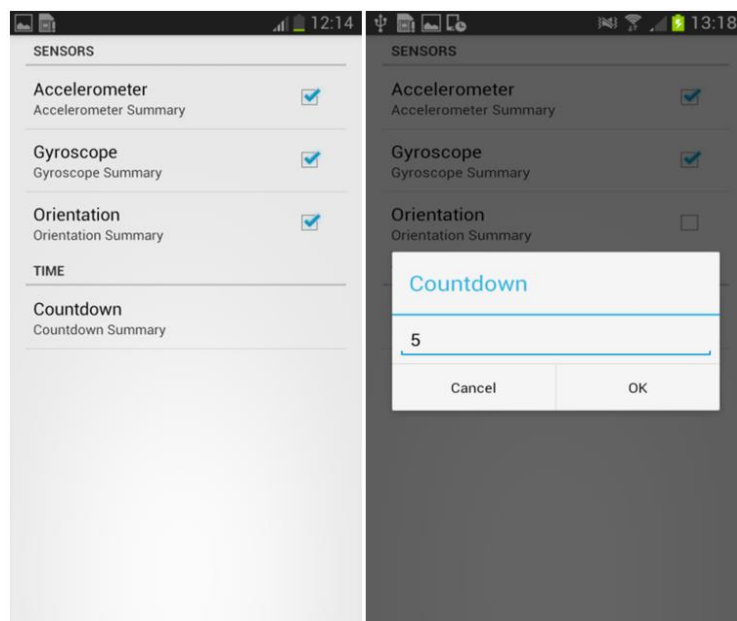
Η εικόνα που ακολουθεί δείχνει την δομή του αρχείου xml όπου περιγράφονται οι ρυθμίσεις του συστήματος. Το αρχικό(root) συστατικό του xml αρχείου είναι το PreferenceScreen. Ένα επίπεδο χαμηλότερα είναι οι 2 κατηγορίες που χωρίζονται οι ρυθμίσεις. Η πρώτη κατηγορία είναι η SensorsCategory και αφορά τους αισθητήρες και η δεύτερη η timeCategory και αφορά τον χρόνο. Στην κατηγορία των αισθητήρων υπάρχουν τρία συστατικά τα οποία είναι accelerometer, gyroscope και orientation. Αυτές οι τρεις επιλογές στην οθόνη των ρυθμίσεων έχουν την μορφή checkbox και ο χρήστης μπορεί να επιλέξει ποιούς αισθητήρες είναι ενεργοί και να καταγράψουν σε αρχεία όταν η εφαρμογή δουλεύει. Οι προεπιλεγμένες τιμές και για τους τρεις αισθητήρες ήταν TRUE. Η κατηγορία του χρόνου περιέχει ένα editTextPreference το οποίο έχει την δυνατότητα να του εισάγουμε μόνο αριθμούς με μέγιστο πλήθος δύο. Αυτή

η ρύθμισή αφορά στο χρόνο σε δευτερόλεπτα που η εφαρμογή θα περιμένει μέχρι να ξεκινήσει την καταγραφή από τη στιγμή που πατάμε το κουμπί που ξεκινάει να καταγράφει. Η ρύθμιση αυτή ήταν πολύ σημαντική κατά τη διάρκεια των καταγραφών αφού μπορούσαμε να μεταβάλλουμε τον χρόνο αυτόν για κάθε εθελοντή ρυθμίζοντας την διάρκεια της αντίστροφης μέτρησης όπως αυτός επιθυμούσε.



Εικόνα 17: Δημιουργία κλιμακίου με τις ρυθμίσεις

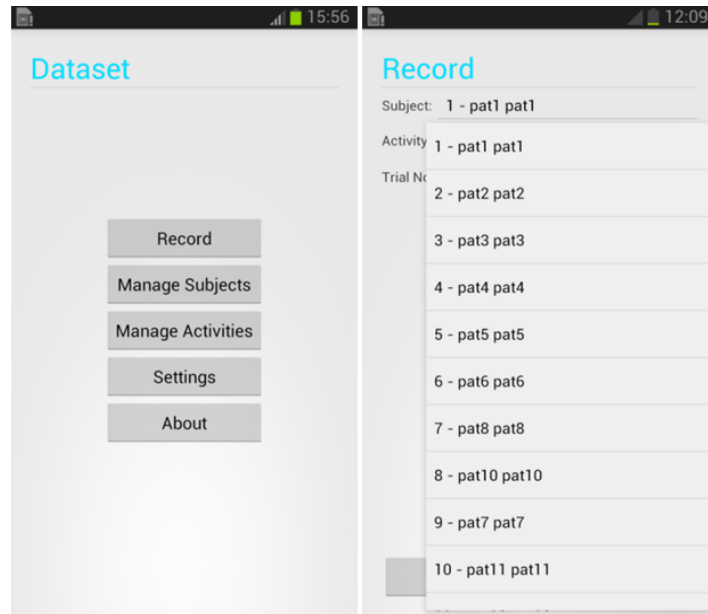
Όταν εκτελείται η εφαρμογή και μεταβούμε στις ρυθμίσεις της θα δούμε την Εικόνα 18α. Εκεί μπορούμε να επιλέξουμε ποιους αισθητήρες θα χρησιμοποιήσουμε στις καταγραφές. Επίσης μπορούμε πατώντας στο countdown να δούμε την Εικόνα 18β όπου μπορούμε να επιλέξουμε τον χρόνο της αντίστροφης μέτρησης σε δευτερόλεπτα μέχρι να ξεκινήσει η καταγραφή.



Εικόνα 18: Ρυθμίσεις εφαρμογής

4.2.6 Παρουσίαση χρήσης εφαρμογής

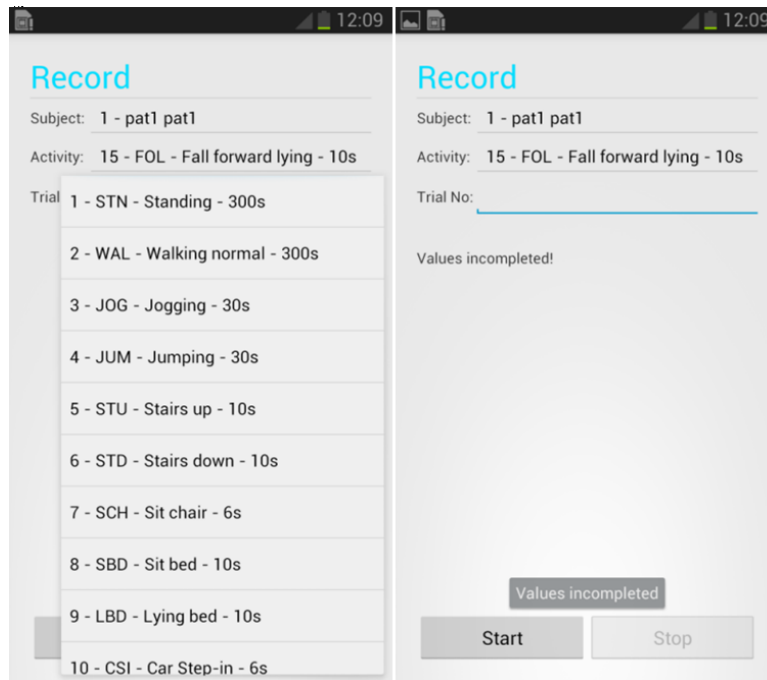
Η εφαρμογή ξεκινάει να τρέχει και εμφανίζει στον χρήστη την αρχική οθόνη στην οποία ο χρήστης μπορεί να επιλέξει μία από τις εξής επιλογές (Εικόνα 19).



Εικόνα 19: Αρχική οθόνη εφαρμογής (α) οθόνη καταγραφής

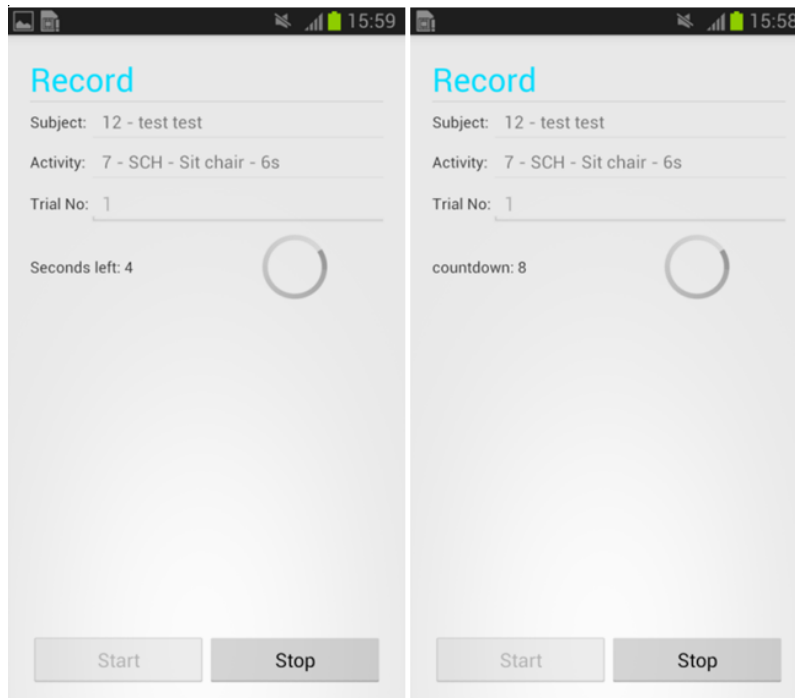
- Record: Μετάβαση στην οθόνη όπου θα γίνει η εγγραφή.
- Manage Subjects: Μετάβαση στην οθόνη που θα μπορεί ο χρήστης να δει και να διαχειριστεί τους εθελοντές.
- Manage Activities: Μετάβαση στην οθόνη που θα μπορεί ο χρήστης να δει και να διαχειριστεί της δραστηριότητες και τις πτώσεις.
- Settings: Μετάβαση στην οθόνη ρυθμίσεων.
- About: Μετάβαση στην οθόνη πληροφοριών της εφαρμογής.

Όταν ο χρήστης πατήσει το record η εφαρμογή θα τον κατευθύνει στην οθόνη που θα γίνει η καταγραφή. Εκεί ο χρήστης θα πρέπει να επιλέξει έναν εθελοντή από το dropdown list (Εικόνα 19β), μια δραστηριότητα από το dropdown list (Εικόνα 20α) και θα πρέπει να συμπληρωθεί και ένας αριθμός δοκιμής(trial). Εάν δεν συμπληρωθούν όλα τα πεδία δεν μπορεί να γίνει η καταγραφή και εμφανίζεται στον χρήστη το μήνυμα ότι τα πεδία δεν είναι συμπληρωμένα(Εικόνα 20β).



Εικόνα 20: Επιλογή εθελοντή και δραστηριότητας πριν την καταγραφή

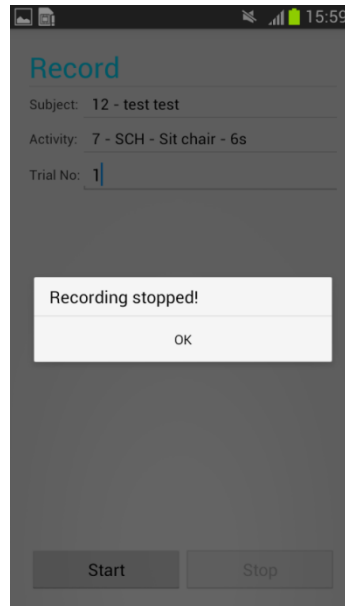
Όταν ο χρήστης συμπληρώσει τα πεδία και πατήσει το κουμπί start ξεκινάει η αντίστροφη μέτρηση και εμφανίζεται στην οθόνη η αντίστροφη μέτρηση(Εικόνα 21). Τον χρόνο που διαρκεί η αντίστροφη μέτρηση μπορούμε να τον ρυθμίσουμε από τις ρυθμίσεις της εφαρμογής.



Εικόνα 21: Καταγραφή σε εξέλιξη

Όταν τελειώσει η αντίστροφη μέτρηση ξεκινάει η καταγραφή. Για να γνωρίζει ο χρήστης ότι η καταγραφή ξεκίνησε, το κινητό εκτελεί μία ηχητική ειδοποίηση. Για όσο η εφαρμογή καταγράφει εμφανίζει πόσα δευτερόλεπτα απομένουν μέχρι να τελειώσει η καταγραφή (Εικόνα 21).

Στη συνέχεια όταν η καταγραφή τελειώσει, και πάλι το κινητό ειδοποιεί ηχητικά τον χρήστη και εμφανίζει ένα παράθυρο διαλόγου (dialog) που αναφέρει ότι η καταγραφή ολοκληρώθηκε (Εικόνα 22).

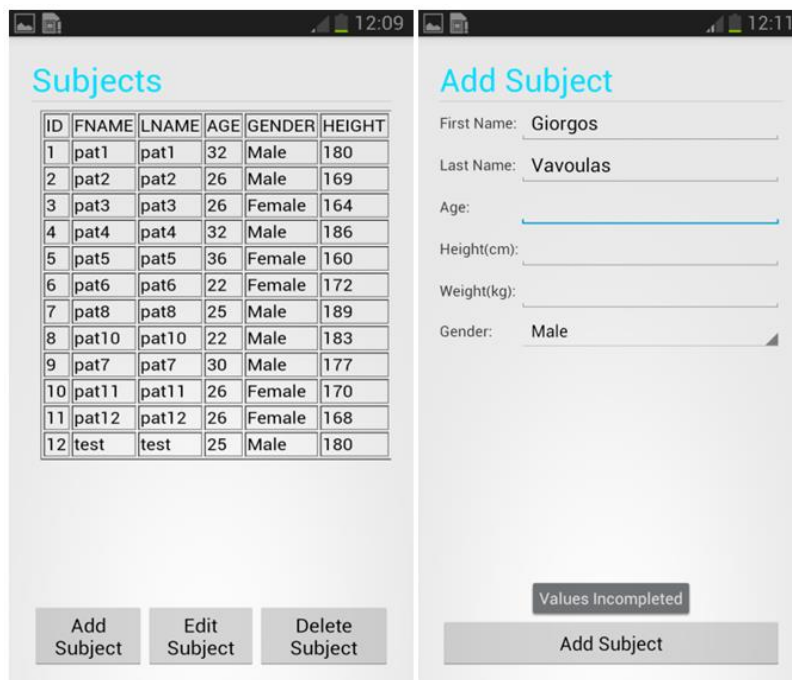


Εικόνα 22: Ολοκλήρωση καταγραφής

Στη συνέχεια αφού ο χρήστης κλείσει αυτό το παράθυρο διαλόγου (dialog) εμφανίζει ένα μήνυμα ότι πρέπει να ελέγξει τον αριθμό δοκιμής (trial number) έτσι ώστε να μην γίνει εγγραφή μίας επόμενης δοκιμής επάνω σε αυτήν που ήδη κατέγραψε.

Τέλος κατά τη διάρκεια της εγγραφής ενεργοποιείται το κουμπί stop το οποίο σταματάει την εγγραφή και επιστρέφει στην αρχική οθόνη της εφαρμογής (kill activity).

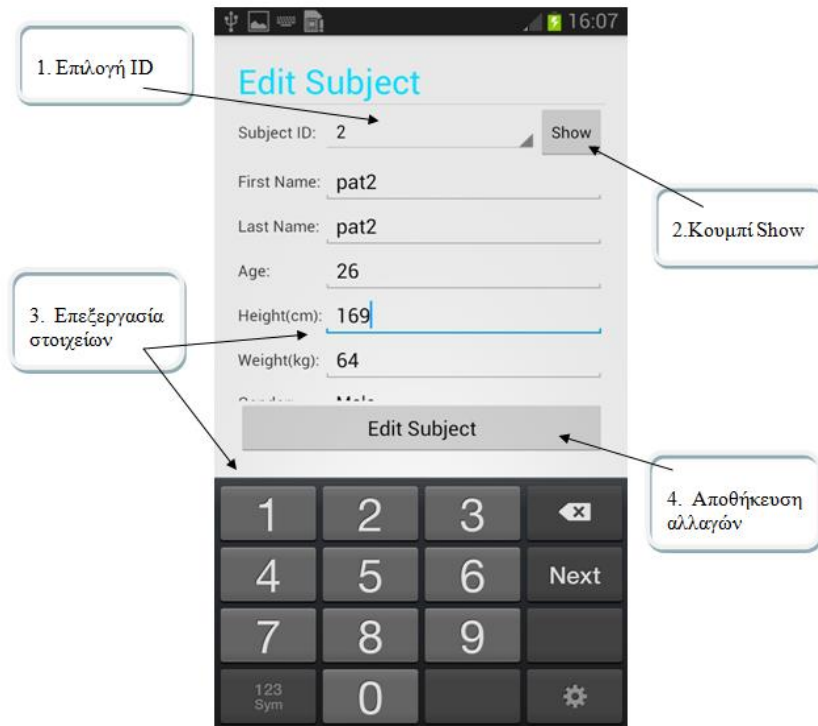
Γυρίζοντας πίσω στην αρχική οθόνη και επιλέγοντας Manage subjects η εφαρμογή θα μας κατευθύνει στην οθόνη όπου μπορούμε να διαχειριστούμε τους εθελοντές μας. Στην οθόνη manage subjects εμφανίζεται ένας πίνακας με όλους τους εθελοντές και τα στοιχεία τους. Κάτω από τον πίνακα υπάρχουν 3 κουμπιά όπου από αυτά ο χρήστης μπορεί να προσθέσει, να επεξεργαστεί ή να διαγράψει κάποιον εθελοντή (Εικόνα 23).



Εικόνα 23: Διαχείριση εθελοντών

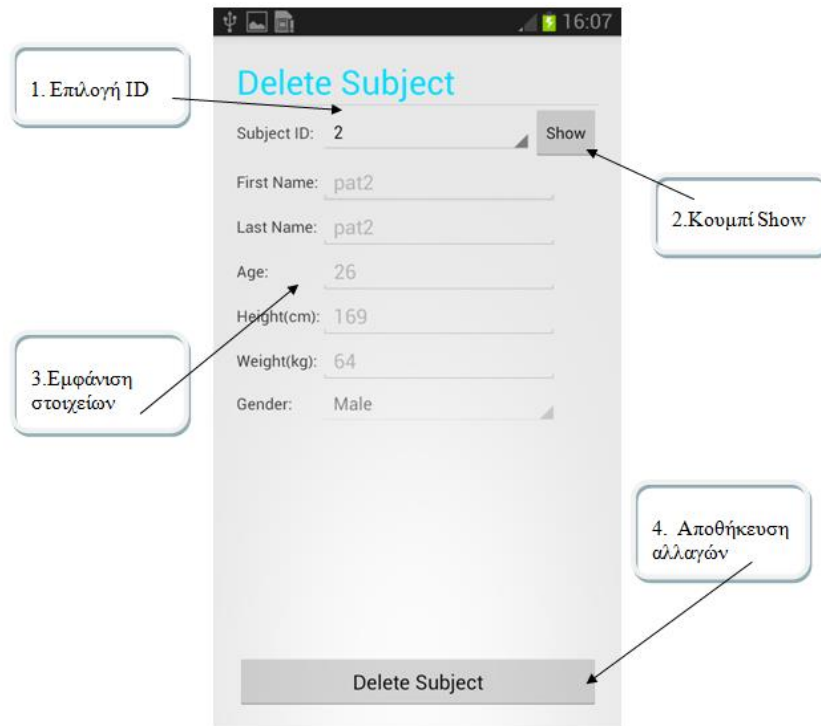
Όταν ο χρήστης πατήσει το κουμπί add subject η εφαρμογή θα τον κατευθύνει στη σελίδα απ όπου μπορούμε να προσθέσουμε έναν εθελοντή. Εκεί και πάλι θα πρέπει να συμπληρωθούν όλα τα πεδία, όνομα, επώνυμο, ηλικία, ύψος και βάρος και από την αναδιπλούμενη λίστα (dropdown list) το φύλο του εθελοντή. Εάν δεν συμπληρωθεί κάποιο από αυτά και πατήσουμε το κουμπί add δεν θα γίνει η καταχώρηση αλλά θα εμφανιστεί ένα μήνυμα στον χρήστη ότι τα πεδία δεν είναι συμπληρωμένα (Εικόνα 23). Όταν συμπληρώσουμε τα πεδία και πατήσουμε το add τότε θα γίνει η εγγραφή στη βάση δεδομένων και θα προσθέσει τον νέο εθελοντή. Εάν πάνε όλα καλά τότε θα εμφανιστεί το μήνυμα ότι ο χρήστης προστέθηκε επιτυχώς. Επιστρέφοντας στην προηγούμενη οθόνη της εφαρμογής (manage subjects) εκεί θα δούμε την νέα καταχώρηση με όλα τα στοιχεία της.

Για την επεξεργασία κάποιου εθελοντή επιλέγουμε από την αναδιπλούμενη λίστα (dropdown list) το id του subject και στη συνέχεια πατάμε το show και εμφανίζονται όλα τα στοιχεία του επιλεγμένου subject. Από εκεί μπορούμε να επεξεργαστούμε τα στοιχεία του subject. Όταν πατήσουμε edit subject θα γίνει η ενημέρωση στον πίνακα των subjects.



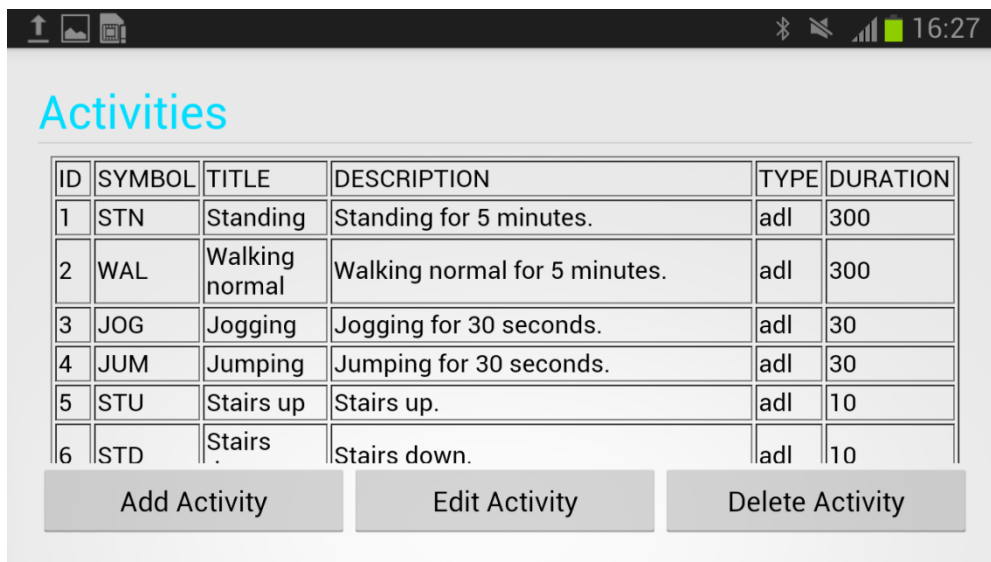
Εικόνα 24: Επεξεργασία εθελοντή

Για την διαγραφή επιλέγουμε, όπως και στην επεξεργασία, το id του εθελοντή και στη συνέχεια πατάμε το show για να μας εμφανίσει τα στοιχεία του. Τα στοιχεία εμφανίζονται στα πεδία αλλά χωρίς την δυνατότητα επεξεργασίας τους. Απλά μόνο για να μπορούμε να τα δούμε. Όταν πατήσουμε το κουμπί για την διαγραφή και ο εθελοντής που επιλέξαμε διεγράφη επιτυχώς εμφανίζεται το μήνυμα στον χρήστη της εφαρμογής ότι ο χρήστης με το id που επιλέξαμε διαγράφηκε επιτυχώς.



Εικόνα 25: Διαγραφή εθελοντή

Στην οθόνη διαχείρισης των δραστηριοτήτων εμφανίζονται όπως και στους εθελοντές όλες οι δραστηριότητες που έχουμε καταχωρημένες στην εφαρμογή (Εικόνα 26).



Εικόνα 26: Διαχείριση δραστηριοτήτων

Με τον ίδιο τρόπο, όπως και στους εθελοντές, λειτουργεί και η διαχείριση των δραστηριοτήτων. Πιο συγκεκριμένα για να προσθέσουμε να επεξεργαστούμε ή να διαγράψουμε κάποια δραστηριότητα ή πτώση χρησιμοποιούμε τα αντίστοιχα κουμπιά που εμφανίζονται στην Εικόνα 26.

4.3 Επεξεργασία δεδομένων

4.3.1 Εισαγωγή αρχείων στο Matlab

Για την επεξεργασία των αρχείων το πρώτο βήμα ήταν να περαστούν τα δεδομένα στο Matlab. Αυτό έγινε με την δημιουργία ενός m-file το οποίο διάβαζε τα .txt αρχεία που περιγράφηκαν στο κεφάλαιο 0 και τα έκανε σε μορφή που μπορούν να διαβαστούν από το Matlab. Τέτοια αρχεία είναι τα mat-files και η κατάληξή τους είναι .mat.

Αρχικά όλα τα αρχεία ταξινομήθηκαν σε φακέλους με βάση το activity code (Πίνακας 3&Πίνακας 4) και δημιουργήθηκαν 13 φάκελοι, όσες δηλαδή οι δραστηριότητες και οι πτώσεις. Αυτό έγινε ώστε το m-file να παίρνει δυναμικά από τους φακέλους τα αρχεία txt. Στη συνέχεια κάθε αρχείο διαβαζόταν από το Matlab και το αποθήκευε σε μία μεταβλητή τύπου Dataset του Matlab. acc για το accelerometer, gyro για το gyroscope, ori για το orientation και τα αποθήκευε σε μορφή .mat.

Έτσι όταν ολοκληρώθηκε αυτή η διαδικασία όλα τα δεδομένα από τα αρχεία txt είχαν μετατραπεί σε αρχεία mat.

4.3.2 Συγχρονισμός αρχείων

Ο συγχρονισμός των αρχείων κρίθηκε απαραίτητος για τον λόγο ότι τα αρχεία είχαν διαφορετικό timestamp και διαφορετική δειγματοληψία. Αυτό συμβαίνει γιατί η συνάρτηση onSensorChanged() του Android τρέχει για όλους τους αισθητήρες όποτε αλλάζει η τιμή του αισθητήρα και όχι ταυτόχρονα για κάθε αισθητήρα. Για τον συγχρονισμό των αρχείων χρησιμοποιήθηκε η τεχνική γραμμικής παρεμβολής (Linear interpolation). Συχνά αυτή η τεχνική χρησιμοποιείται σε σήματα επιτάχυνσης για να πραγματοποιηθεί σταθερή δειγματοληψία. Έτσι λοιπόν υλοποιήθηκε το m-file που θα είχε είσοδο τα 3 mat αρχεία ,πλεον, και θα είχε έξοδο ένα ενοποιημένο αρχείο με 10 στήλες. Οι δέκα στήλες προκύπτουν από το: timestamp, acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z, azimuth, pitch, roll. Οκοινός χρόνος (timestamp) που επιλέχθηκε να συγχρονιστούν όλοι οι αισθητήρες είναι το χρονικό αποτύπωμα (timestamp) του γυροσκοπίου. Επιλέχθηκε το συγκεκριμένο επειδή η δειγματοληψία του γυροσκοπίου ήταν η μεγαλύτερη από τους άλλους τρεις αισθητήρες. Μετά τον συγχρονισμό τοπλήθος των αρχείων μειώθηκαν κατάτο ένα τρίτο.

```
Interpolaccx = interp1(acc.timestamp,acc.x,gyro.timestamp, 'linear');
```

```
Interpolaccy = interp1(acc.timestamp,acc.y,gyro.timestamp, 'linear');
```

```
Interpolaccz = interp1(acc.timestamp,acc.z,gyro.timestamp, 'linear');
```

```
Interpolyaw = interp1(ori.timestamp,ori.yaw,gyro.timestamp, 'linear');
```

```
Interpolpitch = interp1(ori.timestamp,ori.pitch,gyro.timestamp, 'linear');
```

```
Interpolroll =interp1(ori.timestamp,ori.roll,gyro.timestamp, 'linear');
```

4.3.3 Annotation

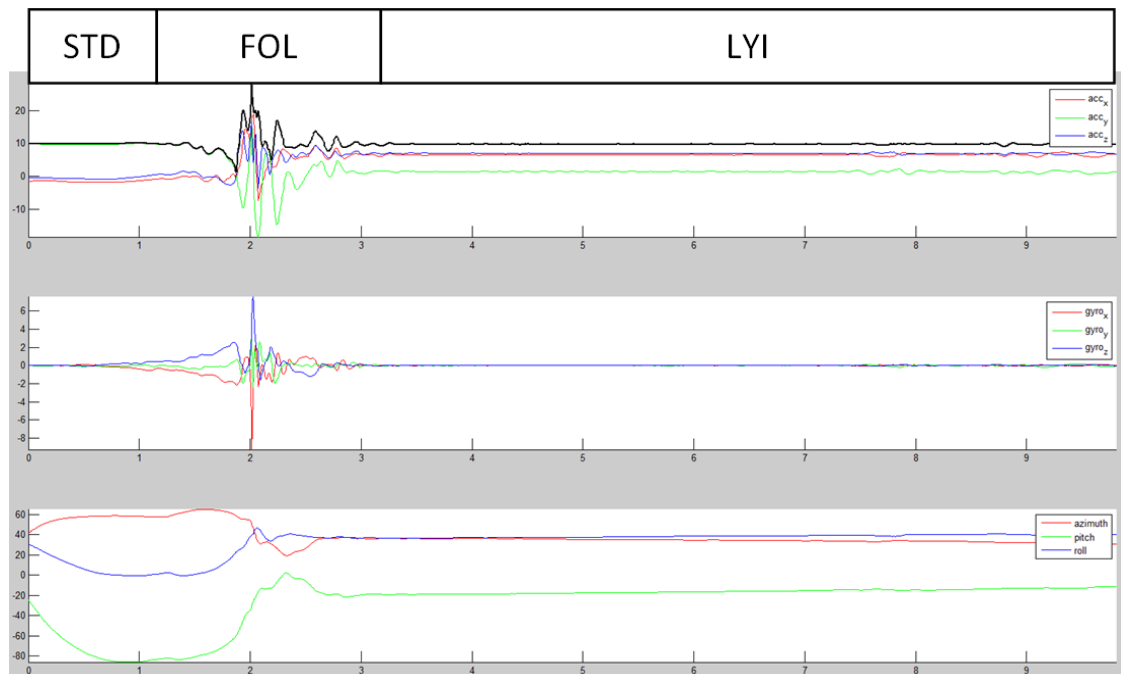
Για να χρησιμοποιηθούν τεχνικές μηχανικής μάθησης (Machine Learning) στο σετ δεδομένων χρειάστηκε επίσης να μπουν ετικέτες στα δεδομένα ώστε να γνωρίζουμε την κίνηση που γίνεται κάθε χρονική στιγμή.

Από την φύση τους τα περισσότερα αρχεία διακρίνονταν σε 3 καταστάσεις. Η πρώτη κατάσταση είναι η αρχική κατάσταση του εθελοντή. Η δεύτερη και η πιο σημαντική είναι η δραστηριότητα που έκανε και η τρίτη είναι η κατάσταση που βρισκόταν ο εθελοντής όταν η καταγραφή τελείωσε. Για παράδειγμα στην πτώση μπροστά ο εθελοντής ξεκινάει την καταγραφή και το σήμα που παίρνουμε στα πρώτα δευτερόλεπτα είναι ότι ο εθελοντής στέκεται (STD), στη συνέχεια γίνεται η πτώση που συνήθως διαρκεί 2-4 δευτερόλεπτα (FOL) και τέλος το σήμα που λαμβάνεται είναι το σήμα που ο εθελοντής βρίσκεται στο στρώμα ξαπλωμένος (LYI). Οι τρεις διακριτές φάσεις ισχύουν για όλες τις καταγραφές εκτός από το standing στο οποίο ο εθελοντής μόνο στεκόταν, το walking που άλλαζε από standing σε walking και η καταγραφή τελείωνε πριν σταματήσει. Το jogging και jumping που επίσης η καταγραφή τελείωνε πριν ο εθελοντής σταματήσει το activity.

Στην Εικόνα 27 φαίνονται οι τρεις διαδοχικές φάσεις της μπροστά πτώσης τύπου FOL. Οι τιμές των χ-αξόνων είναι οι τιμές του relative timestamp με μονάδα μέτρησης τον χρόνο σε δευτερόλεπτα. Από πάνω προς τα κάτω σε κάθε τομέα της εικόνας υπάρχουν τα σήματα της επιτάχυνσης, του γυροσκοπίου και του προσανατολισμού. Στο πρώτο σήμα φαίνεται η επιτάχυνση στους τρεις άξονες x,y,z. Επίσης η μαύρη γραμμή απεικονίζει την συνολική επιτάχυνση του σήματος η οποία υπολογίζεται από τον τύπο:

$$M_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

και χρησιμοποιείται συχνά στην ανίχνευση της πτώσης. Παρατηρώντας την εικόνα βλέπουμε ότι η πτώση έχει μία κορυφή στο σημείο κοντά στο 2 sec. Αυτό το σημείο είναι και το σημείο που έγινε η κρούση με το έδαφος. Μετά από αυτό ακολουθεί η ακινησία με τον χρήστη να παραμένει στο έδαφος.



Εικόνα 27: Annotation

Ο σχολιασμός (annotation) έγινε και αυτός με τη χρήση του Matlab. Κάθε συγχρονισμένο αρχείο άνοιξε και εμφανιζόταν στην οθόνη η γραφική παράστασή του. Σύμφωνα με την δραστηριότητα κάθε αρχείο εμφάνιζε τις παραπάνω διαφορές. Για λόγους ευκολίας

δημιουργήθηκε η βοηθητική σχετική χρονοσφραγίδα (relative timestamp) η οποία μετρούσε τον χρόνο με σημείο έναρξης το μηδέν. Έτσι το m-file που φτιάχτηκε δημιουργούσε μία επιπλέον στήλη στο αρχείο και έβαζε την ετικέτα (label) η οποία δείχνει την δραστηριότητα που γίνεται κάθε χρονική στιγμή, σε κάθε γραμμή του αρχείου.

4.3.4 Τελικά δεδομένα

Μετά το annotationη μορφή των αρχείων άλλαξε. Η πρώτη αλλαγή είναι ότι τα αρχεία τύπου.txtμετατράπηκαν σε αρχεία τύπου .csv (comma separated values). Επίσης άλλαξε και η δομή μέσα στα αρχεία καθώς η πρώτη γραμμή των αρχείων δηλώνει την επικεφαλίδα της κάθε στήλης (όπως ορίζει η μορφοποίηση csv)και προστέθηκε μία επιπλέον στήλη η οποία έχει την ετικέτα (label) κάθε δραστηριότητας. (Βλ. 4.3.3).

```
timestamp,acc_x,acc_y,acc_z,label
2693088902000,0.7182605,9.825804,-1.1109096,STD
2693289745000,0.7565677,9.720459,-1.0247183,STD
2693489138000,0.82360536,9.835381,-0.8714894,STD
2693501929000,0.79487497,9.80665,-0.9864111,STD
2693512780000,0.79487497,9.797073,-1.1013328,STD
2693524453000,0.7757214,9.758766,-1.0534488,STD
2693534002000,0.7757214,9.768343,-1.0821792,STD
2693547322000,0.7565677,9.730036,-1.0630256,STD
2693558625000,0.78529817,9.710882,-1.0342951,STD
2693573179000,0.78529817,9.739613,-0.93852705,STD
2693584137000,0.8331822,9.80665,-0.90021986,STD
2693593722000,0.842759,9.854534,-0.8714894,STD
2693603934000,0.8331822,9.864111,-0.79487497,STD
2693615990000,0.79487497,9.873688,-0.7757214,STD
2693625887000,0.7374141,9.854534,-0.7565677,STD
2693638532000,0.6895301,9.854534,-0.7661445,STD
2693649208000,0.63206923,9.835381,-0.7469909,STD
2693662552000,0.63206923,9.797073,-0.7469909,FKL
2693674207000,0.5363012,9.787497,-0.7469909,FKL
2693685247000,0.51714754,9.768343,-0.7086837,FKL
2693696697000,0.5267244,9.787497,-0.7278373,FKL
2693707910000,0.48841715,9.749189,-0.6991069,FKL
2693717840000,0.4405331,9.720459,-0.6799533,FKL
2693727813000,0.41180268,9.682152,-0.6607997,FKL
2693738320000,0.37349546,9.672575,-0.6607997,FKL
2693750218000,0.36391866,9.672575,-0.7182605,FKL
2693760802000,0.33518824,9.739613,-0.81402856,FKL
2693774643000,0.33518824,9.739613,-0.8714894,FKL
2693787773000,0.31603462,9.787497,-0.9672575,FKL
2693800226000,0.35434186,9.768343,-0.90979666,FKL
```

Εικόνα 28: Τελική μορφοποίηση του αρχείου με δεδομένα επιταχυνσιόμετρου

4.4 Αλγόριθμοι που βασίζονται σε κατωφλίωση

Για την αξιολόγηση των αλγορίθμων σε κινητά τηλέφωνα υλοποιήθηκαν τρεις δημοσιευμένοι αλγόριθμοι ανίχνευσης πτώσης που βασίζονται σε κατωφλίωση (threshold-based algorithms) και έχουν χρησιμοποιηθεί σε κινητά τηλέφωνα. Οι αλγόριθμοι αυτοί χρησιμοποιούν τα δεδομένα από το επιταχυνσιόμετρο για να ανιχνεύσουν τις διαφορετικές φάσεις μίας πτώσης. Η ακολουθία των φάσεων αυτών εξαρτάται από τους αλγόριθμους και μπορεί να είναι:

- Ξεκίνημα της πτώσης,
- ταχύτητα,
- ανίχνευσή κρούσης,
- αλλαγή του προσανατολισμού

- και παρακολούθηση της στάσης του σώματος.

Οι συγκεκριμένοι αλγόριθμοι που υλοποιήθηκαν είναι παρουσιάζονται στην συνέχεια, και οι παράμετροι που αυτοί χρησιμοποιούν εμφανίζονται τον Πίνακα 6.

Method	Parameter	Value
Sposaro and Tyson [1]	TH1 _a	5.5 m/s ²
	TH1 _b	16 m/s ²
	TH1 _c	80°
	TH1 _d	± 10°
	W1 _a	0.5 s
	W1 _b	1 s
	W1 _c ¹	3 s
Dai et al. [6]	TH2 _{a, c, d}	17 m/s ²
	TH2 _b	10 m/s ²
	W2 _{a, c}	20 samples
	W2 _{b, d}	200 samples
He et al. [8]	TH3 _a	12 m/s ²
	TH3 _b	8 m/s ²
	TH3 _c	40°

Πίνακας 7: Παράμετροι που χρησιμοποιούνται από κάθε αλγόριθμο

Αλγόριθμος των Sposaro και Tyson [1].

Ο πρώτος αλγόριθμος είναι των Sposaro και Tyson [1]. Ο αλγόριθμος αυτός χρησιμοποιεί την συνολική επιτάχυνση (πλάτος της επιτάχυνσης) για κάθε δείγμα στον χρόνο:

$$M_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

όπου x,y,z είναι η επιτάχυνση κάθε άξονα. Μια πτώση μπορεί να έχει συμβεί όταν ένα κατώτατο όριο (TH1a) και ένα ανώτατο όριο (TH1b) ξεπεραστούν μέσα σε ένα μικρό χρονικό διάστημα (W1a). Σύμφωνα με τους συγγραφείς αυτά τα όρια μπορούν να παραμετροποιηθούν σύμφωνα με την ηλικία, το ύψος, το βάρος και το επίπεδο της δραστηριότητας. Μετά από αυτή τη φάση, ο αλγόριθμος ελέγχει εάν ο προσανατολισμός της

συσκευής άλλαξε (TH1c). Τέλος, εάν η αλλαγή του προσανατολισμού παραμένει σταθερή (TH1d) για ένα άλλο χρονικό παράθυρο (W1c) τότε ανιχνεύεται η πτώση.

Αλγόριθμος του Dai et al. [6]

Αυτός ο αλγόριθμος χρησιμοποιεί την συνολική επιτάχυνση όπως και ο “ifall” [1], και χρησιμοποιεί επίσης και τα δεδομένα του προσανατολισμού υπολογίζοντας το MV_i. Ο αλγόριθμος αυτός βασίζεται στη διαφορά μεταξύ της μέγιστης και ελάχιστης τιμής μέσα σε ένα χρονικό παράθυρο. Εάν η διαφορά αυτή είναι μεγαλύτερη ενός ορίου τότε ένα χρονικό παράθυρό τρέχει και ελέγχει εάν ένα άλλο όριο έχει ξεπεραστεί κοιτώντας για το longlie. Εάν αυτό το όριο δεν έχει ξεπεραστεί τότε υπάρχει υποψία πτώσης. Το ίδιο γίνεται και για το MV_i. Εάν και οι δύο προϋποθέσεις ισχύουν τότε η πτώση έχει συμβεί.

$$MV_i = x_i \sin \gamma_i + y_i \sin \beta_i - z_i \cos \beta_i \cos \gamma_i$$

Αλγόριθμος του He et al. [6]

Ο τελευταίος αλγόριθμος χρησιμοποιεί μία μέθοδο η οποία αρχικά υπολογίζει την τιμή του signal magnitude area (SMA) για κάθε χρονική στιγμή και η τιμή αυτή όσο μεγαλύτερη είναι τόσο πιο βίαιη είναι η κίνηση που γίνεται. Στη συνέχεια εάν το προκαθορισμένο όριο του SMA (TH3_a) έχει ξεπεραστεί το σύστημα προχωράει στον υπολογισμό του M_i. Εάν η τιμή που υπολογίστηκε είναι μεγαλύτερη από το όριο του M_i (TH3_b) τότε υπολογίζεται γωνία κλίσης (tilt angle - TA) και το αποτέλεσμά του δείχνει εάν ο χρήστης εκτελεί κάποια οριζόντια ή κάθετη δραστηριότητα. Εάν και αυτή η τιμή είναι μεγαλύτερη από το προκαθορισμένο όριο (TH3_c) για το γωνία κλίσης τότε η έξοδος του συστήματος είναι ότι έγινε πτώση.

$$SMA_i = \frac{1}{i} \left(\sum_{u=1}^i |x_u| + \sum_{u=1}^i |y_u| + \sum_{u=1}^i |z_u| \right)$$

$$TA_i = \sin^{-1} \left(\frac{y_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}} \right)$$

4.5 Αλγόριθμοι μηχανικής μάθησης

Για την εξαγωγή των χαρακτηριστικών χρησιμοποιήθηκε ένα κινούμενο παράθυρο (sliding window) δύο δευτερολέπτων με 50% επικάλυψη (overlap). Οι τιμές αυτές επιλέχθηκαν εμπειρικά και είναι μέσα στα όρια της βιβλιογραφίας [10] [16] [28] [29]. Για το κάθε χρονικό παράθυρο υπολογίστηκαν τα χαρακτηριστικά στα σήματα που περιγράφονται στο υποκεφάλαιο 4.5.1. Η εξαγωγή των χαρακτηριστικών έγινε μόνο στο βασικό κομμάτι της κάθε δραστηριότητας όπως περιγράφεται και στο annotation. Για παράδειγμα στα αρχεία FKL έγινε εξαγωγή των χαρακτηριστικών από το σήμα μόνο στο σημείο που το σήμα είχε τιμή FKL.

4.5.1 Εξαγωγή των χαρακτηριστικών

Για κάθε ένα από τους άξονες του επιταχυνσιόμετρου και γυροσκοπίου υπολογίστηκαν τα εξής χαρακτηριστικά:

- Η μέση τιμή του σήματος (Mean)

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- Διάμεσος (Median)
- Τυπική Απόκλιση (Standard deviation): Η διασπορά
- Λοξότητα (skewness)
- Κυρτότητα (Kurtosis)
- Ελάχιστη τιμή (Minimum)
- Μέγιστη τιμή (Maximum)

Για τους τρεις άξονες του επιταχυνσιόμετρου υπολογίστηκε η:

- Κλίση (Slope)

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

Για το tilt angle υπολογίστηκαν τα:

- Η μέση τιμή του σήματος(Mean)
- Διασπορά(Standard deviation)
- Λοξότητα (Skewness)
- Κυρτότητα (Kurtosis)

Για την συνολική επιτάχυνση υπολογίστηκαν τα:

- Η μέση τιμή του σήματος (Mean)
- Η Διασπορά (Standard deviation)
- Λοξότητα (skewness)
- Κυρτότητα (Kurtosis)
- Ελάχιστη τιμή (Minimum)
- Μέγιστη τιμή (Maximum)
- Διαφορά μέγιστης και ελάχιστης τιμής
- Εντροπία της ενέργειας του σήματος (EEB)
- Σύντομος χρόνος ενέργειας (Short time energy)
- Φασματική κατωφλίωση (Spectral roll-off)
- Φασματικό κέντρο βάρους (Spectralcentroid)
- Φασματική ροή (Spectralflux)

4.5.2 Χρήση αλγορίθμων μηχανικής μάθησης

Όλα τα χαρακτηριστικά εισήχθησαν σε αρχεία τύπου .csv με τις τιμές τους.

Αφού ολοκληρώθηκε η διαδικασία εξαγωγής των χαρακτηριστικών του σήματος ακολούθησε η διαδικασία εισαγωγής των μεταβλητών αυτών στο weka. Αρχικά τα δεδομένα (dataset) δομήθηκαν με τέτοιο τρόπο ώστε να είναι εφικτό να γίνει αξιολόγηση. Αρχικά χρησιμοποιήθηκε το ενσωματωμένο φίλτρο του weka για να συγχωνευτούν όλα τα ADL σε μία κλάση με το όνομα “NOFALL”. Αυτό έγινε με σκοπό να ενοποιηθούν όλες οι

δραστηριότητες σε μία κλάση αφού το πρόβλημα που μελετούσαμε ήταν της ανίχνευσης πτώσης και της αναγνώρισης πτώσης. Το φίλτρο που χρησιμοποιήθηκε ήταν:

```
weka.filters.unsupervised.attribute.MergeManyValues-C60-  
LNOFALL-R2,3,6,7,8,10,11,12,13
```

το οποίο συγχωνεύει τις τιμές από τις γραμμές 2,3,6,7,8,10,11,12,13 και τους δίνει την τιμή NOFALL και το αποτέλεσμα μπαίνει στη στήλη 60.

Στην συνέχεια μειώθηκε η συχνότητα δειγματοληψίας ή ρυθμός δειγματοληψίας (subsampling) στα δείγματα που είχαμε στο σύνολο δεδομένων. Αυτό έγινε γιατί υπήρχαν πολλές περιπτώσεις που δεν αντιπροσώπευαν πτώση (NO FALL) και λιγότερες που αντιστοιχούσαν σε πτώση (FALLS). Με στόχο λοιπόν την εξισορρόπηση του αριθμού των περιπτώσεων NOFALL και των FALL χρησιμοποιήθηκε το φίλτρο του

```
weka:weka.filters.supervised.instance.SpreadSubsample-M1.0-  
X0.0-S1.
```

Η αξιολόγηση που αποφασίστηκε να γίνει περιλάμβανε τρεις διαφορετικούς αριθμούς από χαρακτηριστικά (features) και δύο διαφορετικές μεθόδους αξιολόγησης. Σαν αποτέλεσμα δημιουργήθηκαν τα ακόλουθα έξι ξεχωριστά σύνολα:

1. Όλα τα χαρακτηριστικά.
2. 10 επιλεγμένα χαρακτηριστικά: που επιλέχθηκαν μετά από δοκιμές ως τα πλέον κατάλληλα για την ανίχνευση πτώσης. Αυτά ήταν η μέση τιμή για τους άξονες του επιταχυνσιόμετρου και γυροσκοπίου (6 χαρακτηριστικά), η κλίση (slope) (1 χαρακτηριστικό), η εντροπία της ενέργειας του σήματος (EEB) (1 χαρακτηριστικό) και η διαφορά μέγιστης και ελάχιστης τιμής για την συνολική επιτάχυνση (1 χαρακτηριστικό).
3. Αυτόματα επιλεγμένα χαρακτηριστικά με την χρήση της κλάσης *bestfirst*³ του *weka*
 - i. Ανίχνευση πτώσης: μέση τιμή επιτάχυνσης του άξονα (1 χαρακτηριστικό), μέγιστη τιμή επιτάχυνσης του άξονα (1 χαρακτηριστικό), λοξότητα επιτάχυνσης x, y, z άξονων (2 χαρακτηριστικά), κυρτότητα γυροσκοπίου στους y, z άξονες (2 χαρακτηριστικά), κλίση (slope) (1 χαρακτηριστικό), η εντροπία της ενέργειας του σήματος (EEB) (1 χαρακτηριστικό) και spectral flux (1 χαρακτηριστικό).
 - ii. Ταξινόμηση πτώσης: διάμεσος και διασπορά άξονα επιτάχυνσης (2 χαρακτηριστικά), μέση τιμή και ελάχιστη τιμή του y άξονα της επιτάχυνσης (2 χαρακτηριστικά), λοξότητα επιτάχυνσης στον x άξονα (1 χαρακτηριστικό), διάμεσο και λοξότητα γυροσκοπίου στον z άξονα (2 χαρακτηριστικά), μέση τιμή γυροσκοπίου στους άξονες x και y (2 χαρακτηριστικά), κυρτότητα γυροσκοπίου στον x άξονα (1 χαρακτηριστικό), EEB (1 χαρακτηριστικό) και φασματική ροή (Spectral flux) (1 χαρακτηριστικό).

Οι μέθοδοι που επιλέχθηκαν για αξιολόγηση των αλγορίθμων ήταν:

³<http://wiki.pentaho.com/display/DATAMINING/BestFirst>

- a) 10 fold cross validation.
- b) 2/3 train & 1/3 test

Κατά την μέθοδο 10 fold cross validation αυτό που συμβαίνει είναι ότι τα δείγματα με τυχαίο τρόπο χωρίζονται σε 10 υποσύνολα. Κάθε φορά μένει εκτός ένα υποσύνολο για δοκιμή και τα υπόλοιπα χρησιμοποιούνται για εκπαίδευση (training) του αλγορίθμου. Αυτό επαναλαμβάνεται 10 φορές. Το πλεονέκτημα αυτής της μεθόδου είναι ότι όλα τα δεδομένα χρησιμοποιούνται και για την εκπαίδευση του αλγορίθμου αλλά και για την δοκιμή του (testing).

Για την πιο ρεαλιστική αξιολόγηση έγινε διαχωρισμός του συνόλου των δεδομένων (dataset) σε υποσύνολο εκπαίδευσης (training set) και υποσύνολο δοκιμής (test set). Σαν αποτέλεσμα το σύστημα εκπαιδεύεται με το ένα υποσύνολο (training set) και στη συνέχεια χωρίς να ξέρει για τα υπόλοιπα δεδομένα προσπαθεί να βρει την κατηγορία στην οποία ανήκουν τα δεδομένα του άλλου υποσυνόλου (test set). Τα δεδομένα χωρίστηκαν ανά εθελοντή και με τέτοιο τρόπο ώστε να υπάρχουν στην ίδια αναλογία όλες οι δραστηριότητες. Ο διαχωρισμός έγινε ως εξής :

- Υποσύνολο εκπαίδευσης (Training set): 1, 6, 12, 13, 14, 15, 16, 17, 18, 19(μόνο πτώσεις), 2, 3, 4, 5, 7, 8(πτώσεις &ADL).
- Υποσύνολο δοκιμής (Test set): 20, 21, 29, 30, 31(μόνο πτώσεις), 9, 10, 11(πτώσεις &ADL).

Δοκιμάστηκαν και αξιολογήθηκε η συμπεριφορά αρκετών αλγορίθμων ταξινόμησης. Συγκεκριμένα, οι αλγόριθμοι που εκτελέστηκαν ήταν οι : *Naive Bayes*, *J48 decision tree*, *Random Forest decision tree* και *Sequential Minimal Optimization-SVM*. Η αξιολόγηση της συμπεριφοράς όλων αυτών των αλγορίθμων έδειξε ότι ο αλγόριθμος *K-NN* έχει καλύτερη απόδοση. Ο αλγόριθμος *K-NN* είναι ένας απλός αλγόριθμος ταξινόμησης ο οποίος υπολογίζει την απόσταση του στιγμιότυπου που ελέγχει και ταξινομεί το στιγμιότυπο σύμφωνα με την απόσταση των στιγμιότυπων που έχει εκπαιδευτεί. Το *k* ορίζει τον αριθμό των γειτόνων που ελέγχει ο αλγόριθμος για την ταξινόμηση. Ο αριθμός *k* που χρησιμοποιήθηκε στη παρούσα εργασία ήταν 1.

5 Πειραματικά Αποτελέσματα

5.1 Αλγόριθμοι βασισμένοι σε κατώφλια

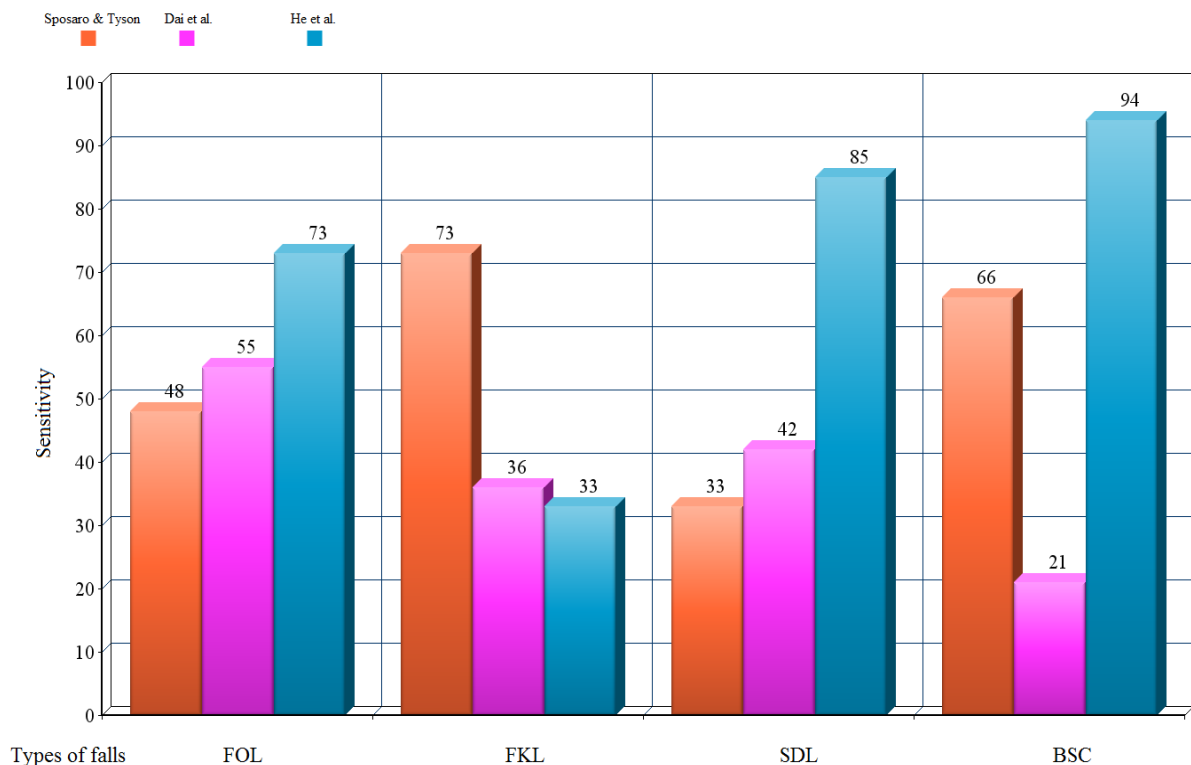
Μετά την υλοποίηση των αλγορίθμων στο Matlab [25] τα αποτελέσματα που πήραμε φαίνονται στον (πίνακα). Σε αυτό το σημείο χρησιμοποιήθηκαν δείγματα πτώσεων από 11 εθελοντές και ADL από 9 εθελοντές.

Τύπος-Δραστηριότητας	Κωδικός	Sposaro and Tyson [1]		Dai et al. [6]		He et al [8]	
		Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
Fall	FOL	0.48	-	0.55	-	0.73	-
	FKL	0.73	-	0.36	-	0.33	-
	SDL	0.33	-	0.42	-	0.85	-
	BSC	0.66	-	0.21	-	0.94	-
ADL	STD	-	1.00	-	1.00	-	1.00
	WAL	-	1.00	-	1.00	-	1.00
	JOG	-	-	-	1.00	-	0.44
	JUM	-	1.00	-	1.00	-	0.74
	STU	-	1.00	-	1.00	-	1.00
	STN	-	1.00	-	1.00	-	1.00
	SCH	-	1.00	-	1.00	-	1.00
	CSI	-	1.00	-	0.96	-	0.63

	CSO	-	1.00	-	1.00	-	0.74
	Total:	0.55	1.00	0.39	1.00	0.71	0.84

Πίνακας 8: Αποτελέσματα αξιολόγησης threshold-based αλγορίθμων

Στον Πίνακα 8 φαίνεται το Sensitivity και το Specificity (Κεφ2.3) του κάθε αλγορίθμου ανά δραστηριότητα.



Εικόνα 29: Σύγκριση του κάθε αλγορίθμου ανάλογα με τον τύπο της πτώσης

Παρατηρείται ότι κάθε αλγόριθμος έχει σημεία υπεροχής του αλλά και τις αδυναμίες του σε σχέση με διαφορετικούς τύπους δραστηριότητες αλλά και πτώσεις. Για παράδειγμα ο αλγόριθμος των *He et al* [8] έχει την καλύτερη συνολική απόδοση αλλά έχει την μικρότερη ευαισθησία (sensitivity) στις πτώσεις FKL, ενώ ο αλγόριθμος των *Sposaro και Tyson* [1] έχει το μεγαλύτερο sensitivity στις πτώσεις τύπου FKL και έχει εν γένει πολύ χαμηλή απόδοση. Η απόδοση αλγορίθμου των *Dai et al.* [6] είναι η χαμηλότερη. Ένας πιθανός λόγος είναι το γεγονός ότι ο αλγόριθμος χρησιμοποιεί και τα δεδομένα από τον αισθητήρα προσανατολισμού, όμως στις καταγραφές που έγιναν για τη δημιουργία του συνόλου των δεδομένων μας ο προσανατολισμός του τηλεφώνου ήταν τυχαίος. Επίσης η γενικά μεγάλη ειδικότητα (specificity) που παρατηρήθηκε μας οδηγεί στο συμπέρασμα ότι οι παράμετροι των αλγορίθμων ήταν αρκετά συντηρητικοί [30].

Η συνολικά χαμηλή ευαισθησία (sensitivity) που παρατηρείται στα αποτελέσματα των τριών αλγόριθμων οφείλεται στους παραμέτρους που χρησιμοποιήθηκαν. Αυτές φαίνονται στον Πίνακα 7, και επιλέχθηκαν με στόχο να βελτιστοποιήσουν την ειδικότητα (specificity).

Ο αλγόριθμος των *Sposaro and Tyson* [1] στην δραστηριότητα τύπου JOG δεν μας έδωσε κάποιο αποτέλεσμα. Αυτό συνέβη γιατί ο χρόνος καταγραφής της δραστηριότητας αυτής στα αρχεία που χρησιμοποιήθηκαν αποδείχτηκε να μην είναι αρκετός. Αυτό είναι και το μεγαλύτερο πρόβλημα στο σύνολο δεδομένων που παρήχθησαν, αφού αρκετές δραστηριότητες και είδη πτώσεων αφαιρέθηκαν και δεν ελέγχθηκαν γι' αυτό τον λόγο.

5.2 Αποτελέσματα ανίχνευσης πτώσης και κατηγοριοποίησης πτώσης με τεχνικές μηχανικής μάθησης

Στον παρακάτω πίνακα παρουσιάζονται τα αποτελέσματα που πήραμε με την χρήση του συνόλου των χαρακτηριστικών που έχουν περιγραφεί στο κεφάλαιο 4.5.1 **Error! Reference source not found.** Η αξιολόγηση των αποτελεσμάτων έγινε σε δύο φάσεις. Στην πρώτη γίνεται με 10-fold cross validation και στην δεύτερη με διαχωρισμό των εθελοντών. Αυτό που περιμέναμε να δούμε από τα αποτελέσματα φαίνεται στην παράμετρο fall classification (Πίνακας 8) με διαχωρισμό των εθελοντών που έχει χαμηλή ακρίβεια (Accuracy) και στις πτώσεις τύπου FOL και SDL και πολύ χαμηλό TPrate.

		Class	sensitivity	FP Rate	Accuracy
Instance-wise 10-fold cross- validation	Fall detection	F	0.994	0.012	99.13%
		NF	0.988	0.006	
	Fall classification	BSC	0.935	0.011	94.65%
		FKL	0.904	0.021	
		FOL	0.825	0.017	
		SDL	0.908	0.010	
		NF	0.991	0.003	
Subject-wise percentage split: 2/3 for training, 1/3 for testing	Fall detection	F	1.000	0.017	99.16%
		NF	0.983	0.000	
	Fall classification	BSC	0.814	0.056	84.45%
		FKL	0.852	0.028	
		FOL	0.550	0.028	
		SDL	0.533	0.067	
		NF	0.983	0.000	

Πίνακας 9: Αποτελέσματα ανίχνευσης και ταξινόμησης πτώσης με χρήση του συνόλου των χαρακτηριστικών

Στα αυτόματα επιλεγμένα χαρακτηριστικά βλέπουμε και πάλι το χαμηλό accuracy στο fall classification στο split για τους ίδιους λόγους όπως και στην πρώτη περίπτωση. Το φίλτρο που χρησιμοποιήθηκε ήταν το: “*BestFirst*” (weka.attributeSelection.BestFirst -D 1 -N 5).

		Class	TP Rate	FP Rate	Accuracy
Instance-wise 10-fold cross- validation	Fall detection	F	0,997	0,017	98,99%
		NF	0,983	0,003	
	Fall classification	BSC	0,902	0,021	91,04%
		FKL	0,767	0,027	
		FOL	0,762	0,025	
		SDL	0,805	0,026	
		NF	0,997	0,003	
Subject-wise percentage split: 2/3 for training, 1/3 for testing	Fall detection	F	1,000	0,017	99,16%
		NF	0,983	0,000	
	Fall classification	BSC	0,814	0,036	86,14%
		FKL	0,667	0,024	
		FOL	0,700	0,032	
		SDL	0,667	0,063	
		NF	1,000	0,008	

Πίνακας 10: Ανίχνευση και ταξινόμηση πτώσης με χρήση των χαρακτηριστικών που επέλεξε το BestFirstφίλτρο

Στα 10 επιλεγμένα χαρακτηριστικά έχουμε 100% ακρίβεια (accuracy) στα αποτελέσματα που αφορούν την ανίχνευση πτώσης όμως παρατηρείται ότι στον τομέα της ταξινόμησης της πτώσης υστερούν σε σχέση με τα αυτόματα επιλεγμένα χαρακτηριστικά και του συνόλου των χαρακτηριστικών. Αυτό το αποτέλεσμα είναι αναμενόμενο αν λάβουμε υπόψη ότι αυτό το σετ χαρακτηριστικών επιλέχθηκε ύστερα από δοκιμές για την μεγιστοποίηση της ακρίβειας στην ανίχνευση πτώσης.

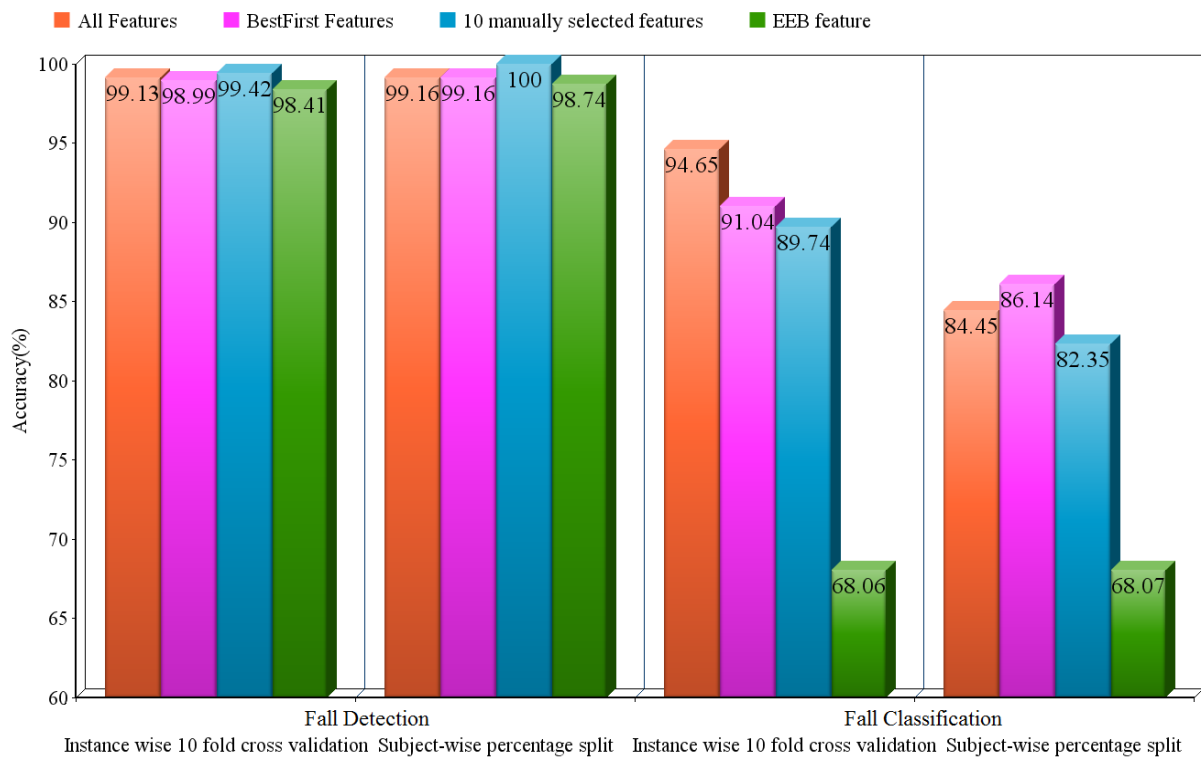
		Class	TP Rate	FP Rate	Accuracy
Instance-wise 10-fold cross- validation	Fall detection	F	0,997	0,009	99,42%
		NF	0,991	0,003	
	Fall classification	BSC	0,878	0,011	89,74%
		FKL	0,685	0,040	
		FOL	0,667	0,040	
		SDL	0,897	0,023	
		NF	0,991	0,003	
Subject-wise percentage split: 2/3 for training, 1/3 for testing	Fall detection	F	1,000	0,000	100%
		NF	1,000	0,000	
	Fall classification	BSC	0,674	0,036	82,35%
		FKL	0,630	0,047	
		FOL	0,650	0,060	
		SDL	0,633	0,058	
		NF	1,000	0,000	

Πίνακας 11: Ανίχνευση και ταξινόμηση πτώσης με χρήση των 10 επιλεγμένων χαρακτηριστικών

Τέλος παρατηρήθηκε ότι το EEB χαρακτηριστικό είναι αρκετά ισχυρό στην ανάλυση των αποτελεσμάτων και δοκιμάσαμε να το αξιολογήσουμε μόνο του. Παρατηρείται ότι μονάχα με αυτό το χαρακτηριστικό τα αποτελέσματα δεν πλησιάζουν σε ακρίβεια τις προηγούμενες περιπτώσεις. Παρόλα αυτά η μεγάλη ακρίβεια που μας δίνει στην ανίχνευση πτώσεων μας δείχνει ότι είναι ένα χαρακτηριστικό που μπορεί να χρησιμοποιηθεί σε αλγόριθμους ανίχνευσης πτώσεων.

		Class	TP Rate	FP Rate	Accuracy
Instance-wise 10-fold cross- validation	Fall detection	F	0,986	0,017	98,41%
		NF	0,983	0,014	
	Fall classification	BSC	0,593	0,105	68,06%
		FKL	0,178	0,076	
		FOL	0,317	0,078	
		SDL	0,287	0,102	
		NF	0,983	0,009	
Subject-wise percentage split: 2/3 for training, 1/3 for testing	Fall detection	F	0,975	0,000	98,74%
		NF	1,000	0,025	
	Fall classification	BSC	0,512	0,087	68,07%
		FKL	0,185	0,081	
		FOL	0,200	0,069	
		SDL	0,433	0,115	
		NF	1,000	0,025	

Πίνακας 12: Αποτελέσματα ανίχνευσης και ταξινόμησης πτώσης με χρήση μόνο του χαρακτηριστικού EEB



Εικόνα 30: Σύγκριση των διαφορετικών σετ χαρακτηριστικών στην αντίγνωση πτώσης και ταξινόμηση πτώσης.

6 Επίλογος

6.1 Συμπεράσματα

Η παρούσα εργασία παρουσιάζει ένα σετ δεδομένων με έναν αρκετά ικανοποιητικό αριθμό από πτώσεις και ADLs. Για την δημιουργία του σχεδιάστηκε και αναπτύχθηκε μία εφαρμογή σε Androidη οποία κατέγραφε δεδομένα από τους αισθητήρες επιτάχυνσης, γυροσκόπιου και προσανατολισμού του κινητού τηλεφώνου. Αυτό το σετ δεδομένων χρησιμοποιήθηκε για την αξιολόγηση τριών αλγορίθμων που χρησιμοποιούν κατώφλια για την ανίχνευση πτώσης και έχουν υλοποιηθεί σε κινητά τηλέφωνα. Επίσης έγινε ανίχνευση και ταξινόμηση πτώσης με χρήση τεχνικών μηχανικής μάθησης.

Τα αποτελέσματα έδειξαν ότι ο κάθε αλγόριθμος που χρησιμοποιεί κατώφλια έχει τα δυνατά του σημεία αλλά και αδυναμίες σε διαφορετικούς τύπους πτώσεων. Με τον αλγόριθμο των He et al. [8] να αποδίδει καλύτερα και με συνολικό sensitivity 0.71. Η αδυναμία του αλγόριθμου αυτού εντοπίζεται στις πτώσεις τύπου FKL όπου και είχε την χαμηλότερη απόδοση συγκριτικά με τους άλλους δύο. Επίσης για το συνολικά χαμηλό sensitivity του αλγορίθμου των Dai et al. [6] ίσως να ευθύνεται η χρήση του αισθητήρα προσανατολισμού στον αλγόριθμο αφού το κινητό τηλέφωνο δεν ήταν τοποθετημένο με συγκεκριμένο προσανατολισμό επάνω στον εθελοντή κατά την διάρκεια των καταγραφών.

Τα αποτελέσματα με χρήση μηχανικής μάθησης έδειξαν αρκετά ικανοποιητικά αποτελέσματα για την ανίχνευση πτώσης. Συγκεκριμένα, με την χρήση του συνόλου των χαρακτηριστικών έχουμε accuracy 99.16% στο σενάριο που έγινε διαχωρισμός των εθελοντών. Το ίδιο αποτέλεσμα είχαμε και με τα αυτόματα επιλεγμένα χαρακτηριστικά. Για την ταξινόμηση πτώσης τα αυτόματα επιλεγμένα χαρακτηριστικά είχαν καλύτερα αποτελέσματα και πάλι στο σενάριο του διαχωρισμού ανά εθελοντή το οποίο είναι περισσότερο ρεαλιστικό από ότι το 10 fold cross validation. Τέλος τα αποτελέσματα με τη χρήση μόνο του EEB χαρακτηριστικού μας δείχνουν ότι είναι ένα χαρακτηριστικό που μπορεί να χρησιμοποιηθεί για την ανίχνευση πτώσης.

Κατά τη διάρκεια της αξιολόγησης των αλγορίθμων που χρησιμοποιούν κατώφλια παρατηρήθηκε ότι στο τέλος ορισμένων δραστηριοτήτων και πτώσεων δεν υπήρχε αρκετός χρόνος για την μελέτη του χρόνου αδράνειας της δραστηριότητας. Αυτό είχε ως αποτέλεσμα στον να μην υπολογιστεί ένας αριθμός από αρχεία στα τελικά αποτελέσματα.

Οι δύο μεγάλες προκλήσεις στον τομέα τις ανίχνευσης της πτώσης με χρήση κινητών συσκευών είναι ότι ο αριθμός των δεδομένων πραγματικών πτώσεων που υπάρχουν είναι εξαιρετικά μικρός και δεν είναι διαθέσιμος αλλά και ότι χρησιμοποιώντας κινητά τηλέφωνα έχουμε περιορισμένη ενέργεια μπαταρίας. Έτσι λοιπόν οδηγούμαστε στην μελέτη του προβλήματος με προσομοιωμένα δεδομένα. Επίσης, όπως φάνηκε και στα αποτελέσματα, οι αλγόριθμοι μηχανικής μάθησης φαίνεται να αποδίδουν καλύτερα από τους αλγόριθμους βασισμένους σε κατώφλια όμως οι αλγόριθμοι αυτοί είναι αρκετά πιο πολύπλοκοι και έτσι χρησιμοποιούν περισσότερους υπολογιστικούς πόρους. Έχοντας υπόψη λοιπόν ότι ένα σύστημα ανίχνευσης πτώσης σε κινητά τηλέφωνα δεν μπορεί να διαθέτει τεράστιο αριθμό υπολογιστικών πόρων με δεδομένο την περιορισμένη ενέργεια της μπαταρίας κάνει απαραίτητη την εύρεση τεχνικών που θα μειώσουν την απαιτούμενη ενέργεια του συστήματος.

6.2 Μελλοντικές Επεκτάσεις

Επέκταση Dataset

Στην μορφή που είναι τώρα το σετ δεδομένων έχει ADL από 9 εθελοντές και πτώσεις από 24 εθελοντές. Το σετ δεδομένων μπορεί να επεκταθεί παίρνοντας δείγματα από περισσότερους εθελοντές. Επίσης μπορούν να γίνουν καταγραφές περισσότερων δραστηριοτήτων και πτώσεων με την ίδια εφαρμογή, απλά προσθέτοντας νέους χρήστες και δραστηριότητες.

Το πρόβλημα που παρουσιάζεται σε ορισμένα activity όπως το jogging και το jumping με το περίοδο αδράνειας στο τέλος, το οποίο δεν ήταν αρκετό για να μελετήσουμε τους αλγόριθμους βασισμένους σε κατώφλια μπορεί να διορθωθεί αντιγράφοντας τα τελευταία δευτερόλεπτα της δραστηριότητας μεγάλωνοντας έτσι τον χρόνο που διαρκεί η κάθε δραστηριότητα.

Αναγνώριση κινήσεων

Το σύνολο δεδομένων που δημιουργήθηκε (Dataset) περιέχει αρχεία όχι μόνο από πτώσεις αλλά και από καθημερινές δραστηριότητες (ADL). Αυτό μας δίνει την δυνατότητα εκτός από το να μελετήσουμε το πρόβλημα της πτώσης, να μελετήσουμε και προβλήματα που αφορούν στην αναγνώριση άλλων δραστηριοτήτων της καθημερινότητας με χρήση κινητών τηλεφώνων.

Υπάρχουν πολλές πιθανές εφαρμογές που μπορούν να αναπτυχθούν αξιοποιώντας την με ακρίβεια αναγνώριση και κατηγοριοποίηση ανθρώπινων δραστηριοτήτων της καθημερινότητας (Human Activity Recognition). Τέτοιες εφαρμογές μπορεί να αφορούν περιβάλλοντα έξυπνων σπιτιών (Smarthouses), σε συστήματα επιτήρησης και παρακολούθησης, διαδραστικές διεπαφές σε παιχνίδια και υπηρεσίες κινητής τηλεφωνίας. Στον τομέα της υγείας για την ενδονοσοκομειακή και εξωνοσοκομειακή παρακολούθηση της κατάστασης των πολιτών/ασθενών.

Αξιολόγηση νέων αλγορίθμων

Οι τρεις αλγόριθμοι που υλοποιήθηκαν και αξιολογήθηκαν στη παρούσα εργασία είναι οι αλγόριθμοι που έχουν χρησιμοποιηθεί σε κινητά τηλέφωνα με την χρήση των ενσωματωμένων αισθητήρων για την ανίχνευση πτώσης. Με το σύνολο δεδομένων (dataset) που δημιουργήθηκε μπορούμε όχι μόνο να υλοποιήσουμε και να αξιολογήσουμε επιπλέον αλγόριθμους αλλά και να αναπτύξουμε νέους αλγόριθμους, το οποίο αποτελεί μελλοντικό στόχο.

Βιβλιογραφία

- [1] F. Sposaro, G. Tyson, "iFall: an Android application for fall monitoring and response," in *31st International Conference IEEE EMBS*, Mineapolis, Minasotta, USA, 2-6 September 2009.
- [2] Marco Avvenuti, Paolo Corsini, Janet Light and Alessio Vecchio Stefano Abbate, "Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network: a Survey," *Wireless Sensor Networks: Application - Centric Design*, 2010.
- [3] Carlos Medrano, Inmaculada Plaza Raul Igual, "Challenges, issues and trends in fall detection," *Biomedical Engineering online*, vol. 12, no. 1, 2013.
- [4] (2014, Jan.) [Online]. <http://www.cs.ucr.edu/~vahid/assistivemonitoring/>
- [5] R. Y. W. Lee, A. J. Carlisle, "Detection of falls using accelerometers and mobile phone technology," *Age and Ageing*, vol. 40, pp. 690–696, 2011.
- [6] J. Dai, X. Bai, Z. Yang, Z. Shen, D. Xuan, "Mobile phone-based pervasive fall detection," *Pers. Ubiquit. Comput.*, vol. 14, pp. 633-643, 2010.
- [7] M. Tolkiehn , L. Atallah , B. Lo , G.-z. Yang, "Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor," in *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 369-372.
- [8] Y. He, Y. Li, C. Yin, "Falling-incident detection and alarm by smartphone with multimedia messaging service (MMS)," *E-Health Telec. Syst. Netw.*, vol. 1, pp. 1-5, 2012.
- [9] S. H.Fang, Y. C.Liang, K. M. Chiu, "Developing a mobile phone-based fall detection system on android platform," in *Communications and Applications Conference*, Hong Kong, 11-13 January 2012.
- [10] Y. Cao, Y. Yang, W. Liu, "E-FallID: A fall detection system using android-based smartphone," in *9th International Conference on Fuzzy Systems and Knowledge Discovery*, Sichuan, 29-31 May 2012, pp. 1509 – 1513.

- [11] V. Q. Viet, G. Lee, D. Choi, "Fall Detection Based on Movement and Smart Phone Technology," in *IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future*, Ho Chi Minh, 27 Feb. - 1 March 2012.
- [12] T. Zhang, J. Wang, P. Liu, J. Hou, "Fall Detection by Embedding an Accelerometer in Cellphone and Using KFD Algorithm," *International Journal of Computer Science and Network Security*, vol. 6, no. 10, pp. 277-284, 2006.
- [13] M. Luštrek, B. Kaluža, "Fall Detection and Activity Recognition with Machine Learning," *International Journal of Computing and Informatics*, vol. 33, no. 2, pp. 197-204, 2009.
- [14] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, A. Vecchio, "A smartphone-based fall detection system," *Pervasive and Mobile Computing*, vol. 8, no. 6, pp. 883–899, 2012.
- [15] Albert MV, Kording K, Herrmann M, Jayaraman A, "Fall Classification by Machine Learning Using Mobile Phones.," *PLoS ONE* 7(5): e36556, 2012.
- [16] Z. Zhaoa, Y.Chena, S. Wanga, Z. Chena, "FallAlarm: Smart Phone Based Fall Detecting and Positioning System," *Procedia Computer Science*, vol. 10, pp. 617-624, 2012.
- [17] P. N. A.Fahmi, V. Q. Viet, D. C. Jai, "Semi-supervised fall detection algorithm using fall indicators in smartphone," in *6th International Conference on Ubiquitous Information Management and Communication*, Kuala Lumpur, Malaysia, 2012.
- [18] A. O. Kansiz, M. A. Guvensan, H. I.Turkmen, "Selection of Time-Domain Features for Fall Detection Based on Supervised Learning," *Lecture Notes in Engineering and Computer Science*, vol. 2208, no. 1, pp. 796-801, 2013.
- [19] A. K. Bourke, J.V. O'Brien, G.M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & Posture*, vol. 26, pp. 194–199, 2007.
- [20] (2014, Jan.) Samsung. [Online]. <http://www.samsung.com/global/galaxys3/>
- [21] (2014, Jan.) STMicroelectronics. [Online]. http://www.st.com/web/catalog/sense_power/FM89/SC1448/PF252427

- [22] (2014, Jan.) Android Developers. [Online].
http://developer.android.com/guide/topics/sensors/sensors_position.html
- [23] (2014, Jan.) wikipedia. [Online].
http://en.wikipedia.org/wiki/Android_%28operating_system%29
- [24] (2014, January) www.tutorialspoint.com. [Online].
http://www.tutorialspoint.com/android/android_architecture.htm
- [25] (2014, Jan.) Mathworks. [Online]. <http://www.mathworks.com/products/matlab/>
- [26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA data mining software: an update," *CM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, 2009.
- [27] (2014, Jan.) android developers. [Online].
<http://developer.android.com/guide/components/fundamentals.html>
- [28] Y. Xuel, L. Jinl, "A naturalistic 3D acceleration-based activity dataset & benchmark evaluations," in *IEEE International Conference on Systems Man and Cybernetics* , 2010, pp. 4081-4085.
- [29] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, "Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine," *International Workshop of Ambient Assisted Living* , vol. 7657, pp. 216-223, 2012.
- [30] G. Vavoulas, M. Padiaditis, E. G. Spanakis, M. Tsiknakis, "The MobiFall Dataset: An Initial Evaluation of Fall Detection Algorithms Using Smartphones," in *6th IEEE International Symposium on Monitoring & Surveillance Research (ISMSR): Healthcare-Bioinformatics*, Chania, Greece, 2013.
- [31] K. Ouchi and M. Doi , "Indoor-outdoor activity recognition by a smartphone," in *ACM Conference on Ubiquitous Computing*, New York, USA, 2012.
- [32] C. McCall, K. Reddy, M. Shah, "Macro-Class Selection for Hierarchical K-NN Classification of Inertial Sensor Data," in *Second International Conference on Pervasive and Embedded Computing and Communication Systems*, Rome, Italy, 2012.

Παράρτημα Α- Ακρωνύμια

ADL	Activity of Daily Living
ADT	Android Development Tools
API	Application Programming Interface
BSC	Back-Sitting Chair
CDMA	Code Division Multiple Access
CSI	Car-Step In
CSO	Car-Step Out
DVM	Dalvik Virtual Machine
ECG	Electrocardiogram
EDGE	Enhanced Data rates for GSM Evolution
EEG	Electroencephalogram
EV-DO	Evolution-Data Optimized
FI	Fall Index
FKL	Forward – Knees- Lying (fall)
FN	False Negative
FOL	FORward - Lying (fall)
FP	False Positives
GPS	Global Position System
GSM	Global System for Mobile Communication
JOG	JOGging
JUM	JUMping
JVM	Java Virtual Machine
KNN	K Nearest Neighbor
MBR	Minimum bounding rectangle (video-based fall detection feature)
MMS	Multimedia Messaging Service

RLR	Regularized Logistic Regression
ROC	Receiver Operating Characteristics
SCH	Sitting Chair
SDL	SiDeward- Lying (fall)
STD	STanDing
SMA	Signal Magnitude Area
SMS	Short Message Service
STN	STairs downN
STU	STairs Up
SVM	Support Vector Machines
TA	Tilt Angle
TN	True Negative
TP	True Positive
UMTS	Universal Mobile Telecommunication systems
VM	Virtual Machine
WAL	WALking
WEKA	Waikato Environment for Knowledge Analysis