



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ
ΒΙΟΙΑΤΡΙΚΗ

**Αναγνώριση πτώσεων με τη χρήση
αισθητήρων "έξυπνων" κινητών τηλεφώνων**

Αριστοτέλης Δήμου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων

Πλαγιανάκος Βασίλης

Λαμία, 2020



**UNIVERSITY OF THESSALY
SCHOOL OF SCIENCE
INFORMATICS AND COMPUTATIONAL
BIOMEDICINE**

Fall detection using "smartphone" sensors

Aristotelis Dimou

**Master Thesis
Supervisor
Plagianakos Vassilis**

Lamia, year



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ
ΒΙΟΙΑΤΡΙΚΗ
ΥΠΟΛΟΓΙΣΤΙΚΗ ΙΑΤΡΙΚΗ ΚΑΙ ΒΙΟΛΟΓΙΑ

**Αναγνώριση πτώσεων με τη χρήση
αισθητήρων "έξυπνων" κινητών τηλεφώνων**

Αριστοτέλης Δήμου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Επιβλέπων
Πλαγιανάκος Βασίλης

Λαμία, 2020

Λαμία, 2020

”Υπεύθυνη δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης”

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [”Αναγνώριση πτώσεων με τη χρήση αισθητήρων ”έξ-υπνων” κινητών τηλεφώνων”] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

Αριστοτέλης Δήμου

Ημερομηνία

26/02/2020

Υπογραφή

**Αναγνώριση πτώσεων με τη
χρήση αισθητήρων "έξυπνων"
κινητών τηλεφώνων**

Αριστοτέλης Δήμου

Τριμελής Επιτροπή:

Πλαγιανάκος Βασίλης, Καθηγητής

Μπάγκος Παντελής, Καθηγητής

Τασουλής Σωτήριος, Επίκουρος Καθηγητής

Λαμία, 2020

Περιεχόμενα

Πρόλογος	8
Περίληψη	9
Abstract	10
1 Εισαγωγή	11
2 Σχετική Έρευνα	13
2.1 Φορετοί αισθητήρες	13
2.1.1 Εργασίες βασισμένες σε κάποιο κατώφλι.	14
2.1.2 Εργασίες βασισμένες σε μεθόδους μηχανικής μάθησης	14
2.2 Αισθητήρες στον περιβάλλοντα χώρο	15
2.2.1 Αισθητήρες εικόνας	15
2.2.2 Άλλοι αισθητήρες	17
3 Μεθοδολογία	18
3.1 Ανάλυση Κύριων Συνιστωσών	18
3.2 Διαδοχική Ανάλυση Κύριων Συνιστωσών χωρίς μητρώο Συνδιακύμανσης . .	19
3.3 Σωρευτικό Αθροιστικό Διάγραμμα Ελέγχου	21
4 Εφαρμογή	23
4.1 Γλώσσα Προγραμματισμού	23
4.2 Αρχική Οθόνη Εφαρμογής	23
4.3 Συλλογή Δεδομένων	25
4.4 Ανάλυση των δεδομένων σε πραγματικό χρόνο	25
5 Αποτελέσματα και Συμπεράσματα	27
5.1 Υλικό	27

5.2	Ανάλυση των αποτελεσμάτων	27
5.3	Κατανάλωση μπαταρίας	30
6	Συζήτηση και μελλοντική μελέτη	32
7	Βιβλιογραφία	34
8	Παράρτημα	41
8.1	Συμπληρωματικά Διαγράμματα	41
8.2	Κώδικα εφαρμογής	41

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του διατμηματικού μεταπτυχιακού κύκλου σπουδών του Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική της Σχολής Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας κατά τη διάρκεια του ακαδημαϊκού έτους 2019-2020, υπό την επίβλεψη του κυρίου Πλαγιανάκου Βασίλη, Καθηγητή του Τμήματος και Πρόεδρο του Μεταπτυχιακού.

Οφείλω ένα μεγάλο ευχαριστώ στους ανθρώπους που με βοήθησαν και με στήριξαν κατά τη διάρκεια των σπουδών μου αλλά και στην ολοκλήρωση της παρούσας εργασίας.

Πιο συγκεκριμένα στον καθηγητή μου Πλαγιανάκο Βασίλειο, που με το θέμα που επιλέξαμε με βοήθησε να εμπλουτίσω τις γνώσεις μου σε ένα πεδίο της επιστήμης που δεν είχα εμβαθύνει αλλά και να αποκτήσω εφόδια για την αγορά εργασίας, εμπλουτίζοντας τις γνώσεις μου σχετικά με τον προγραμματισμό για το λειτουργικό σύστημα Android. Θα ήθελα επίσης να ευχαριστήσω τους φίλους μου Σπύρο Γεωργακόπουλο και Καλαγάνη Φώτιο, για την πολύτιμη βοήθειά τους στα τεχνικά θέματα και στις υπολογιστικές μεθόδους που χρειάστηκαν για την ολοκλήρωση αυτής της πτυχιακής.

Περίληψη

Εδώ και πολλά, πλέον, χρόνια τα κινητά τηλέφωνα έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητάς μας και η ημερήσια χρήση τους όλο και αυξάνεται. Μια συσκευή που παρουσιάζει ολοένα και μεγαλύτερη υιοθέτηση από τον κόσμο [1, 2], ακόμα και στις αναπτυσσόμενες όπως και τις χώρες του τρίτου κόσμου [3]. Το να εξετάσουμε αν αυτή η τάση είναι καλή ή κακή είναι πέρα από το σκοπό της παρούσας εργασίας. Αντικείμενο αυτής αποτελεί το πως μπορούμε να χρησιμοποιήσουμε τα κινητά τηλέφωνα για την βελτίωση της ποιότητας της ζωής μας.

Με τη βελτίωση της ιατροφαρμακευτικής επιστήμης, έχουμε αύξηση στο προσδόκιμο όριο ζωής. Τα άτομα που διανύουν την τρίτη ηλικία είναι περισσότερα, κατ' επέκταση περισσότεροι εξ' αυτών είναι ικανοί να μένουν μόνοι τους στα σπίτια τους και να διατηρούν την αυτονομία τους. Ένας λόγος τραυματισμού σοβαρού ή μη είναι και οι πιθανές πτώσεις που μπορεί να τύχουν σε αυτά τα άτομα, κάτι που όντας μόνοι τους μπορεί να αποβεί μοιραίο.

Στην παρούσα εργασία ασχοληθήκαμε με την συγγραφή μιας εφαρμογής στο λειτουργικό σύστημα Android που θα μπορεί να επισημάνει τις πτώσεις του χρήστη της και κατόπιν απουσίας περαιτέρω ενεργειών από μέρους του, θα επικοινωνεί με κάποιον τρίτο ενημερώνοντας τον ότι το άτομο χρήζει βοήθειας. Σκοπός είναι λοιπόν να μειωθεί ο χρόνος απόκρισης και παροχής ιατρικής βοήθειας σε άτομα που δεν δύνανται να τη ζητήσουν έπειτα από μια πτώση με τη χρήση των ενσωματωμένων αισθητήρων που βρίσκονται στην ίδια την κινητή συσκευή.

Abstract

In recent years, mobile phones have been a ubiquitous and indispensable part of our daily lives, with an ever increasing usage duration but also a growing adoption world wide [1, 2], even in developing and third world countries [3]. We won't stick to if this trend is beneficial or not but in finding a way to use that trend to our advantage and improve our quality of life.

With the advancements in healthcare and pharmaceuticals, we now have a greater life expectancy, therefore more and more of the elderly are living by themselves. A major reason for trauma, severe or not, in that age group is the potential fall that may occur, which by living alone may prove to be even fatal.

In this dissertation we are going to program an application for the Android operating system that will be able to detect when the user is falling and absent further action from his part, will communicate with a third party making known that the user needs immediate care. The purpose of this application is to reduce the response time of medical assistance to people that are unable to ask for it after a fall using only the embedded sensors of the mobile device.

Στη μοντέρνα εποχή της πληροφορίας και της τεχνολογίας, το κινητό τηλέφωνο είναι ένα ευρέως διαδεδομένο μέσο που μας βοηθάει στην καθημερινότητα μας, είτε μειώνοντας τις αποστάσεις με τα αγαπημένα μας πρόσωπα μέσω της επικοινωνίας, είτε παρέχοντας μας πληροφορίες με τη χρήση του διαδικτύου, ψυχαγωγία ακόμη και εργαλείο δουλειάς. Το κινητό τηλέφωνο έχει εξελιχθεί πλέον από απλή συσκευή επικοινωνίας σε συσκευή εντοπισμού, φωτογραφική μηχανή αλλά και το βασικότερο σε ολόκληρο προσωπικό υπολογιστή. Ένα βασικό εξάρτημα των κινητών τα τελευταία χρόνια είναι οι αισθητήρες που ενσωματώνουν, κατά κύριο λόγο αισθητήρες κίνησης αλλά και πλειάδα άλλων. Η πλειοψηφία των “έξυπνων” κινητών έχει τουλάχιστον ένα αισθητήρα επιτάχυνσης και ένα γυροσκόπιο, για να εξυπηρετεί τις βασικές ανάγκες του ίδιου του λειτουργικού του συστήματος σχετικά με τον προσανατολισμό της συσκευής για την διευκόλυνση του γραφικού του περιβάλλοντος. Αυτούς τους αισθητήρες όμως μπορούμε να τους εκμεταλλευτούμε και σε συνδυασμό με την όλο και μεγαλύτερη επεξεργαστική ισχύ και φορητότητα που έχουν αυτές οι συσκευές, να συλλέξουμε αλλά και να επεξεργαστούμε αυτά τα δεδομένα ώστε να μας βοηθήσουν σε πολλές πτυχές της ζωής μας.

Μια εφαρμογή που μπορούν να χρησιμοποιηθούν οι αισθητήρες αυτοί, αλλά και άλλοι ή σε συνδυασμό με άλλες συσκευές που φοράμε, όπως “έξυπνα” ρολόγια ή fitness trackers, είναι η αναγνώριση της δραστηριότητας. Το κινητό μας να μετρά, δηλαδή, τη διάρκεια, την ποιότητα αλλά και άλλα δεδομένα σχετικά με τη δραστηριότητα του χρήστη του, πχ. τρέξιμο, ποιότητα ύπνου, αρτηριακή πίεση κλπ. Ανάμεσα στις ποικίλες εφαρμογές, ιδιαίτερη σημασία έχει η αναγνώριση της πτώσης, με υπολογισμούς που γίνονται στο ίδιο το τηλέφωνο και μπορούν να τρέχουν στο παρασκήνιο συνέχεια χωρίς σημαντικές επιπτώσεις στη διάρκεια ζωής της μπαταρίας αλλά και χωρίς εμφανή επίπτωση στην αποκρισιμότητα της συσκευής. Επιπλέον όλη αυτή η διαδικασία θα πρέπει να γίνεται χωρίς να υπάρχει κάποια ταξινόμηση προϋπάρχουσας πληροφορίας και επιπλέον να δουλεύει σωστά για τον κάθε χρήστη. Ο λόγος που έχουμε θέσει αυτήν την προϋπόθεση είναι ότι, βασικό χαρακτηριστικό της εφαρμογής είναι η επεκτασιμότητα τόσο στις δραστηριότητες που θα μπορεί να ανιχνεύει αλλά και στη βάση χρηστών που θα την χρησιμοποιούν. Θέλουμε λοιπόν η ανάλυση των δεδομένων να

γίνεται στη συσκευή με όσο το δυνατόν λιγότερες διαμορφωμένες τιμές εκ των προτέρων για να μην περιορίζεται από συγκεκριμένες εργαστηριακές δοκιμές.

Μια τέτοια εφαρμογή θα είχε χρησιμότητα σε πολλούς ανθρώπους με έμφαση τους ανθρώπους της τρίτης ηλικίας οι οποίοι κατά κύριο λόγο ζούνε μόνοι τους. Η πρόσφατη βιβλιογραφία μας λέει ότι οι πτώσεις των ηλικιωμένων είναι η έκκτη πιο πιθανή αιτία θανάτου στις ηλικίες 65 και άνω [4]. Επίσης ένας στους τρεις άνω των 65 ετών θα πέσει τουλάχιστον μια φορά το χρόνο [5], κάτι το οποίο στις ηλικίες άνω των 80 αγγίζει ένα ποσοστό της τάξης του 50% [6]. Οι επιπτώσεις που έχει ένα τέτοιο συμβάν στον ενήλικα πέρα από τις προφανείς του τραυματισμού, είναι και ο φόβος του πεσίματος που έχει άμεσο αντίκτυπο στην κοινωνική του ζωή, στην ανεξαρτησία του και τις δραστηριότητες του [7, 8].

Το κινητό είναι μια συσκευή που ακόμα και όταν δεν τη χρησιμοποιούμε είναι μόνιμα πάνω μας και οι αισθητήρες συνεχίζουν να συλλέγουν δεδομένα στο παρασκήνιο. Είναι προφανές λοιπόν η χρησιμότητα μιας εφαρμογής που θα αναγνωρίζει τις πτώσεις κατά τη διάρκεια που συμβαίνουν, σε σχεδόν πραγματικό χρόνο. Επιπλέον θα λειτουργεί μόνο με τη χρήση του κινητού τηλεφώνου, χωρίς πολύπλοκες διαδικασίες και χωρίς επιπλέον συσκευές που θα επιφέρουν δυσκολία, πιθανώς δυσχέρεια αλλά και κόστος στον ηλικιωμένο. Δηλαδή τα δεδομένα θα συλλέγονται και θα επεξεργάζονται από το ίδιο το τηλέφωνο χωρίς την απαίτηση άλλης συσκευής, είτε για τη συλλογή δεδομένων είτε για την ανάλυση τους. Αφότου αναγνωριστεί πτώση και εφόσον ο ηλικιωμένος δεν είναι σε θέση να επικοινωνήσει ο ίδιος με κάποιον στην περίπτωση που χρειάζεται βοήθεια, τότε το ίδιο το τηλέφωνο θα είναι σε θέση αυτόματα να πραγματοποιεί κλήσεις, να αποστέλλει μηνύματα ηλεκτρονικού ταχυδρομείου ή γραπτά μηνύματα καθώς επίσης και συνδυασμό των ανωτέρω. Η επικοινωνία θα γίνεται με επαφή που έχει οριστεί εκ των προτέρων, είτε αυτός είναι κάποιος οικείος, κάποιος αριθμός έκτακτης ανάγκης ή ο γιατρός του ηλικιωμένου. Έτσι η επέμβαση της άμεσης βοήθειας θα είναι όσο αμεσότερη γίνεται με σκοπό την καλύτερη και αποτελεσματικότερη αντιμετώπιση του περιστατικού. Με αυτή την εφαρμογή θα βελτιωθεί η ποιότητα ζωής των ηλικιωμένων, πέρα από το προφανές της σωματικής αλλά και της ψυχικής τους υγείας. Όπως έχει διαπιστωθεί ο φόβος μην τους συμβεί κάποια πτώση επηρεάζει τον τρόπο ζωής τους δυσμενώς, όσον αφορά τις δραστηριότητες τους καθώς και την ανεξαρτησία τους [9].

Αυτή η πτυχιακή είναι βασισμένη σε προηγούμενη εργασία [10], προσπαθώντας να την επεκτείνουμε και να εξετάσουμε αν μπορεί να επιτευχθεί εξολοκλήρου στην ίδια συσκευή με αποδεκτή ακρίβεια στην πράξη αλλά και χωρίς προϋπάρχοντα δεδομένα όπως και η ανάπτυξη της συγκεκριμένης εφαρμογής.

Σχετική Έρευνα

Εξαιτίας της μεγάλης ζήτησης για την εύρεση μιας προσιτής και ταυτόχρονα ικανοποιητικής λύσης στο πρόβλημα της αυτόματης αναγνώρισης πτώσεων, χρόνια τώρα έχουν προταθεί και ερευνηθεί διάφορες μεθοδολογίες. Σε αυτό το κεφάλαιο θα σχολιάσουμε μερικές από αυτές.

Συγκεκριμένα μπορούμε να κατηγοριοποιήσουμε τις διάφορες μεθόδους σε δύο κατηγορίες βάσει του τρόπου λειτουργίας τους.

1. Φορητοί αισθητήρες
2. Αισθητήρες στον περιβάλλοντα χώρο

2.1 Φορητοί αισθητήρες

Κατά κύριο λόγο μιλάμε για αισθητήρες επιτάχυνσης και σε μερικές περιπτώσεις σε συνδυασμό με γυροσκόπια. Αυτοί οι αισθητήρες λόγω του μικρού μεγέθους τους, της διαθεσιμότητας και της προσιτής τιμής τους έχουν χρησιμοποιηθεί εκτενώς στην έρευνα για την αναγνώριση πτώσεων και δραστηριότητας. Οι αισθητήρες αυτοί μπορούν να φορευθούν από το χρήστη κάτω, πάνω ή μαζί με το ρουχισμό του. Σε αυτή την κατηγορία οι μέθοδοι που χρησιμοποιούνται είναι βασισμένοι ή σε κάποιο όριο, στις οποίες σημαίνει πτώση όταν η επιτάχυνση περάσει κάποιο “κατώφλι” ή σε μεθόδους μηχανικής μάθησης. Οι περισσότερες έρευνες χρησιμοποιούν μεθόδους με κατώφλια [11, 12, 13, 14, 15, 16]. Οι αισθητήρες είναι τοποθετημένοι κατά κύριο λόγο στη μέση των χρηστών αλλά και στο αυτί, κορμό, μηρό, πόδι, κεφάλι, καρπό, λαιμό. Σχετικά με τη τοποθεσία του αισθητήρα η μέση προτείνεται γιατί βρίσκεται κοντά στο κέντρο βάρους του σώματος και παρέχει περισσότερο αξιόπιστες πληροφορίες σχετικά με τις κινήσεις του χρήστη. Συνήθως ελέγχουν την ταχύτητα, την πρόσκρουση και την στάση του σώματος μετά αυτήν. Μια υποομάδα αυτής της κατηγορίας είναι οι αισθητήρες που βρίσκονται σε “έξυπνα” κινητά τηλέφωνα. Συνήθως δουλεύουν με χαμηλής πολυπλοκότητας αλγορίθμους βασισμένους στα όρια και λιγότεροι είναι αυτοί που προσπαθούν να εισάγουν τη μηχανική μάθηση. Παρακάτω γίνεται μια συνοπτική παράθεση εργασιών χωρίς να είναι πλήρης.

2.1.1 Εργασίες βασισμένες σε κάποιο κατώφλι.

Almeida et al [17] ενσωμάτωσε γυροσκόπιο σε μπαστούνι που μετράει την γωνιακή ταχύτητα και βάση ενός ορίου αναγνωρίζει την πτώση. Bianchi et al [18] πρόσθεσε βαρόμετρο στους αισθητήρες επιτάχυνσης και το γυροσκόπιο έτσι ώστε να μειώσει τις λανθασμένες πτώσεις που σημαίνει ο αλγόριθμος. Chen et al [13] χρησιμοποιώντας αισθητήρα επιτάχυνσης στην ζώνη του χρήστη και όταν εντοπίσει πτώση με ένα σύστημα ασύρματης μετάδοσης στέλνει σήμα σε κάποιο κεντρικό υπολογιστή για περαιτέρω ενέργειες. Estudilo-Valderrama et al [19] χρησιμοποιεί αισθητήρα επιτάχυνσης στην πλάτη πάνω από τον κόκκυγα και η ανάλυση των δεδομένων γίνεται απομακρυσμένα σε άλλη συσκευή. Kang et al [20] μια συσκευή που φοριέται στον καρπό και έχει ένα αισθητήρα επιτάχυνσης δύο αξόνων και ένα γυροσκόπιο για την εκτίμηση των πτώσεων καθώς και άλλους αισθητήρες ιατρικής φύσεως όπως ηλεκτροκαρδιογράφο, αρτηριακής πίεσης, αναπνευστικού ρυθμού, θερμοκρασίας και παλμικής οξυμετρίας. Lin et al [21] χρησιμοποιεί πλήθος αισθητήρων υδραργύρου ενσωματωμένους σε ρουχισμό (μπλούζα, παντελόνι) που δείχνουν την κλίση των επιμέρους σημείων του σώματος και οπτικούς αισθητήρες που αναγνωρίζουν αν το σώμα του χρήστη βρίσκεται σε οριζόντια θέση. Luo et al [22] ένας αισθητήρας επιτάχυνσης στη μέση και αναπαράσταση σε ένα τρισδιάστατο μοντέλο που συσχετίζει διάφορες στάσεις του σώματος με τις τιμές που δίνει ο αισθητήρας. Mathie et al [23] χρησιμοποιεί αισθητήρες επιτάχυνσης με όρια για τον εντοπισμό δραστηριότητας και πτώσεων αλλά και για την ανάλυση του βαδίσματος. Narayanan et al [24] αισθητήρας επιτάχυνσης στη μέση του ατόμου και μετάδοση της πληροφορίας σε μια πλατφόρμα στον ιστό όπου μπορεί να γίνει παρακολούθηση και αξιολόγηση από ιατρικό προσωπικό. Noury et al [25] δυο αισθητήρες επιτάχυνσης τοποθετημένους ορθογώνια μεταξύ τους, στη μασχάλη του χρήστη. Tamura et al [26] έφτιαξε ένα φορητό αερόσακο που σε συνδυασμό με αισθητήρα επιτάχυνσης και γυροσκόπιο μετρώντας την επιτάχυνση και τη γωνιακή ταχύτητα μπόρεσε να εντοπίσει την πτώση 300 milisecond νωρίτερα από την πρόσκρουση. Wang et al [27] ένας αισθητήρας επιτάχυνσης στο κεφάλι μετράει την επιτάχυνση σε όλους τους άξονες καθώς και την ταχύτητα για να διαφοροποιήσει τις πτώσεις από καθημερινές δραστηριότητες. Williams et al [28] χρησιμοποιεί πιεζοηλεκτρικό αισθητήρα για να μετρήσει την ένταση μιας πρόσκρουσης και με ένα αισθητήρα κλίσης υδραργύρου ελέγχει αν ο χρήστης είναι σε οριζόντια θέση ή απλά χτύπησε κάποιο εμπόδιο. Wu [29] μέτρησε την ταχύτητα σε οριζόντιο και σε κάθετο επίπεδο του κορμού του χρήστη για διάφορες δραστηριότητες και πτώσεις.

2.1.2 Εργασίες βασισμένες σε μεθόδους μηχανικής μάθησης

Doukas et al [30] χρησιμοποίησε αισθητήρα επιτάχυνσης στο πόδι γιατί η πλειοψηφία της ανθρώπινης κίνησης περιλαμβάνει κίνηση του ποδιού σε έναν από τους τρεις άξονες. Τα δεδομένα αποστέλλονται σε άλλη συσκευή που κάνει την ταξινόμηση με χρήση Μηχανών Διανυσμάτων Υποστήριξης (SVM Support Vector Machines). Zhang et al [31] χρησι-

μπούησε δεδομένα από αισθητήρα επιτάχυνσης στη ζώνη του ατόμου για να εξάγει δεδομένα για το μέγεθος και τη χρονική διάρκεια της επιτάχυνσης και τα ταξινομεί με χρήση Μηχανών Διανυσμάτων Υποστήριξης.

Οι φορετοί αισθητήρες, έχουν κατακλύσει την έρευνα τα τελευταία χρόνια κυρίως λόγω του μικρού κόστους και μεγέθους τους αλλά και τη μικρή πολυπλοκότητα στην τοποθέτηση και τη λειτουργία τους. Από την άλλη πλευρά οι περισσότεροι αισθητήρες δουλεύουν έχοντας ένα σταθερό σημείο αναφοράς όσον αφορά τον προσανατολισμό πάνω στο σώμα του χρήστη, συνθήκη που είναι πολύ εύκολο να διακοπεί αλλά επιπλέον είναι παρεμβατικοί και δεν είναι λίγες οι φορές που τα ίδια τα άτομα είτε άθελα τους είτε ηθελημένα αμελούν τη χρήση τους.

2.2 Αισθητήρες στον περιβάλλοντα χώρο

Αυτή η κατηγορία μπορεί να διαιρεθεί σε δύο υποκατηγορίες. Αν ο αισθητήρας είναι εικόνας (κάμερα) ή άλλος αισθητήρας (υπέρυθρης ακτινοβολίας, επιτάχυνσης, μικρόφωνο).

2.2.1 Αισθητήρες εικόνας

Τα τελευταία χρόνια είτε για λόγους ασφαλείας αλλά και την δημιουργία ενός πιο αυτοματοποιημένου τρόπου ζωής, συνηθίζεται η ύπαρξη κάμερας/ών στους χώρους του σπιτιού. Πέρα από τις άλλες λειτουργίες που εξυπηρετούν αυτοί οι αισθητήρες εικόνας, ο εντοπισμός πτώσεων είναι ένας βασικός που έχει μελετηθεί εκτενώς.

Οι έρευνες που έχουν γίνει στο θέμα θα μπορούσαν να κατηγοριοποιηθούν επιμέρους σύμφωνα με τον τρόπο που εντοπίζουν τις πτώσεις. Δηλαδή εντοπισμός ακινησίας, ανάλυση της αλλαγής του σχήματος του σώματος και ανάλυση της κίνησης του κεφαλιού στον τρισδιάστατο χώρο. Παρακάτω παρατίθεται συνοπτικά μια λίστα εργασιών για την καλύτερη εκτίμηση της υπάρχουσας έρευνας.

Εργασίες που κάνουν χρήση κατωφλιών

Lee et al [32] χρησιμοποιεί τη στάση του σώματος και τον προσανατολισμό σε συνδυασμό με τον ορισμό εκ των προτέρων ζωνών ακινησίας ή μειωμένης κίνησης για τον εντοπισμό της πτώσης. Miaou et al [33] με κάμερα τοποθετημένη στην οροφή του δωματίου παίρνει την αναλογία του πλάτους προς το ύψος που αλλάζει από όρθια σε οριζόντια θέση και σε συνδυασμό με πληροφορίες για το κάθε άτομο ξεχωριστά όπως BMI (Body Mass Index) αλλάζει τα όρια για τον εντοπισμό των πτώσεων. Williams et al [34] χρησιμοποιεί πληθώρα αισθητήρων εικόνας χαμηλής ανάλυσης και υπολογιστικής ισχύς, ο εντοπισμός της πτώσης γίνεται σε ξεχωριστή συσκευή (προσωπικό υπολογιστή) και η τοποθέτηση της πτώσης γίνεται χρησιμοποιώντας μια προ βαθμονομημένη κάμερα από το σύνολο των υπαρχόντων. Lin et al [35, 36] εντοπίζει τις πτώσεις χρησιμοποιώντας δισδιάστατες αναπαραστάσεις ατόμων από συμπιεσμένο βίντεο. Tzeng et al [37] χρησιμοποιεί υπέρυθρη κάμερα για να μην επηρεάζεται

από σημεία που έχουν εμπόδια καθώς και αισθητήρες πίεσης στο πάτωμα. Diraco et al [38] χρησιμοποιεί βαθμονομημένη κάμερα και τη σιλουέτα του ατόμου και το κέντρο μάζας του, όταν αυτό είναι πιο κοντά από κάποιο όριο σε σχέση με το πάτωμα τότε σημαίνει πτώση. Mastorakis et al [39] χρησιμοποιώντας τον αισθητήρα του Kinect βρίσκει το πλάτος, ύψος και το βάθος του σώματος μετά με την ταχύτητα και στη συνέχεια με ακινησία εντοπίζει τις πτώσεις. Rougier et al [40, 41] χρησιμοποιεί την κίνηση του σώματος που είναι μεγάλη κατά τη διάρκεια μιας πτώσης, το σχήμα του σώματος και την απουσία κίνησης μετά την πτώση. Σε άλλη έρευνα χρησιμοποιεί την τροχιά που κάνει το κεφάλι του ατόμου κατά την πτώση κοιτάζοντας την ταχύτητα στο οριζόντιο και στο κάθετο επίπεδο. Vishwakarma et al [42] βρίσκει την αναλογία ύψους πλάτους του ατόμου και την γωνία που σχηματίζει μια κάθετη γραμμή που περνάει από το κέντρο του σώματος σε σχέση με την οριζόντια γραμμή του εγγεγραμμένου κουτιού από την αναγνώριση του σώματος. Fu et al [43] κάμερα που μετράει την αλλαγή της αντίθεσης με την αλλαγή της φωτεινότητας στο εκάστοτε πίξελ για τον εντοπισμό των πτώσεων. Delibasis and Maglogiannis [44] χρησιμοποιεί υπερευρυγώνιο φακό (fish-eye) για την εύρεση της μεγαλύτερης ακμής στο αντικείμενο και αν είναι αρκετό το μήκος της και δεν έχει διεύθυνση προς το κέντρο του πεδίου θέασης της κάμερας σημαίνει πτώση.

Εργασίες που χρησιμοποιούν μεθόδους μηχανικής μάθησης

Nait Charif et al [45] χρησιμοποιεί υπερευρυγώνιους φακούς τοποθετημένους στην οροφή που καταγράφουν την κίνηση στα επιμέρους δωμάτια, έπειτα κατηγοριοποιεί τις δραστηριότητες βάσεις των κινήσεων του ατόμου αλλά και το γενικό πλαίσιο και όταν παρατηρηθεί ακινησία σε συγκεκριμένο πλαίσιο τότε εντοπίζει πτώση. Κάνει χρήση μέγιστης εκ των υστέρων πιθανότητας (Maximum a Posteriori). Toreyn et al [46] χρησιμοποιεί Κρυφά Μαρκοβιανά Μοντέλα (HMM Hidden Markov Models) στο ύψος και το πλάτος του εγγεγραμμένου κουτιού του σώματος για να εντοπίσει την πτώση. Επίσης πάλι με HMM σε ηχητικά δεδομένα αυτή τη φορά προσπαθεί να ξεχωρίσει την πτώση από το κάθισμα ή το σκύψιμο. Anderson et al [47] αφού γίνει ο προσδιορισμός του ατόμου από τον περιβάλλοντα χώρο, με HMM στην αναλογία πλάτος προς ύψος εντοπίζει τις πτώσεις. Thome et al [48] χρησιμοποιεί Ιεραρχικά Κρυφά Μαρκοβιανά Μοντέλα (Hierarchical HMM) στην ανθρώπινη φιγούρα για τη στάση του σώματος και τον προσανατολισμό του στο χώρο. Cuchiara et al [49] χρησιμοποιεί περισσότερες βαθμονομημένες κάμερες για μια αναπαράσταση του σώματος σε περισσότερα από ένα δωμάτια ή χώρους με εμπόδια, κάνει χρήση HMM για την ταξινόμηση. Jansen and Deklerek [50] χρησιμοποίησε στερεοσκοπική κάμερα σε γωνία δωματίου για να έχει τη μεγαλύτερη κάλυψη του χώρου και με την πληροφορία του βάθους μέτρησε την ακινησία υπό συγκεκριμένα πλαίσια για να σημαίνει πτώση. Hsu et al [51] εντοπίζει τις πτώσεις κάνοντας χρήση μεθόδου τριγωνοποίησης για να βρει τη στάση του σώματος και προσπαθεί να βρει το μοτίβο σε προϋπάρχοντα δεδομένα από βίντεο ανάλογα με την δραστηριότητα. Hazellhoff et al [52] χρησιμοποιεί δυο κάμερες κάθετες μεταξύ τους και με ανάλυση κυρίων συνισ-

ωσών βρίσκει τον κύριο άξονα στο άτομο καθώς και το λόγο των διακυμάνσεων στις χ και ψ κατευθύνσεις σε συνδυασμό με έλεγχο της κίνησης του κεφαλιού αυξάνει την απόδοση, κάνει χρήση Γκαουσιανής ταξινόμησης. Liu et al [53] χρησιμοποιεί κάμερες και βρίσκει τη σιλουέτα του ατόμου για την προστασία της ιδιωτικότητας και με χρήση του πλάτους και του ύψους ταξινομεί σε πτώση ή όχι με τη χρήση της μεθόδου του κοντινότερου γείτονα. Zhang et al [54] χρησιμοποιεί κάμερα που πληροφορεί και για το βάθος και βάσει της θέσης κεντρικών συνδέσμων του σώματος βρίσκει τις πτώσεις, κάνει χρήση Μηχανών Διανυσμάτων Υποστήριξης.

Είναι προφανές ότι όλες αυτές οι εργασίες έχουν εκπονηθεί για να ξεπεράσουμε τις αδυναμίες που παρουσιάζουν οι διαφορετικοί τρόποι που επιτρέπουν την αναγνώριση των πτώσεων με σκοπό να αυξηθεί η αποτελεσματικότητά τους. Αυτές που βασίζονται στην ακινησία εμφανίζουν συχνότερα λανθασμένες πτώσεις αλλά και καθυστερούν στον εντοπισμό. Αυτές που βασίζονται στην κίνηση του κεφαλιού δείχνουν να τα πηγαίνουν καλύτερα γιατί υπάρχει μεγαλύτερος συσχετισμός στην κίνηση του κεφαλιού σε σχέση με το υπόλοιπο σώμα σε μια πτώση αλλά είναι πιο πολύπλοκος ο εντοπισμός του από μια μόνο κάμερα.

Ένα άλλο πρόβλημα στη χρήση αισθητήρων εικόνας είναι το πρόβλημα της ιδιωτικότητας. Πόσοι θα αισθανόντουσαν άνετα με το να υπάρχουν κάμερες να τους παρακολουθούν καθόλη τη διάρκεια της ημέρας, από την άλλη πλευρά, σε περίπτωση πτώσης μπορεί να επιβεβαιωθεί οπτικά μια πτώση από κάποιον υπάλληλο σε κέντρο άμεσης επέμβασης μέσω μιας ροής βίντεο.

2.2.2 Άλλοι αισθητήρες

Ως άλλους αισθητήρες, θα αναφερθούμε σε αισθητήρες που δεν έχουν να κάνουν με τη χρήση εικόνας για τον εντοπισμό των πτώσεων. Στην κατηγορία αυτή έχουμε τη μικρότερη βιβλιογραφία. Alwan et al [55] χρησιμοποίησε αισθητήρες πίεσης στο έδαφος για να εντοπίσει τις πτώσεις γιατί έχουν διαφορετικό “αποτύπωμα” από τις καθημερινές εργασίες. Sixsmith et al [56, 57] χρησιμοποιεί μια συστάδα υπέρυθρων αισθητήρων για να ξεχωρίσει τα ανθρώπινα σώματα από το υπόβαθρο βάσει της θερμοκρασίας του σώματος και μετά ελέγχει για την ταχύτητα στο κάθετο επίπεδο αλλά και την ακινησία του ατόμου για τον εντοπισμό της πτώσης. Li et al [58] χρησιμοποιεί μια συστάδα μικροφώνων για τον εντοπισμό των πτώσεων.

3.1 Ανάλυση Κύριων Συνιστωσών

Τα τελευταία χρόνια η πρόσβαση σε πληροφορία είναι πιο εύκολη από ποτέ, είναι φανερό λοιπόν ότι τα δεδομένα που συλλέγονται για ανάλυση είναι πολυδιάστατα. Η ανάλυση κυρίων συνιστωσών είναι μια στατιστική μέθοδος που απώτερος σκοπός της είναι η μείωση διάστασης (dimension reduction). Με σκοπό η διάσταση των προς επεξεργασία δεδομένων να είναι μικρότερη χωρίς να χαθεί καθόλου ή έστω ελάχιστη πληροφορία από τα αρχικά δεδομένα. Γεγονός που μειώνει και την απαίτηση σε μνήμη αλλά και την ταχύτερη ανάλυση τους. Αυτό γίνεται χρησιμοποιώντας ένα ορθογώνιο μετασχηματισμό, μετατρέπει δηλαδή τα πιθανώς συσχετισμένα χαρακτηριστικά διανύσματα των παρατηρήσεων σε νέα, γραμμικώς ανεξάρτητα που ονομάζουμε κύριες συνιστώσες. Οι κύριες συνιστώσες που παίρνουμε είναι όσες οι αρχικές μας παρατηρήσεις και τις ταξινομούμε με βάση τη διακύμανση σε φθίνουσα σειρά, δηλαδή η πρώτη από αυτές αντιπροσωπεύει την πληροφορία από το μεγαλύτερο πλήθος των παρατηρήσεων. Αρχικά η μέθοδος αυτή επινοήθηκε από τον Karl Pearson [59] αλλά είναι γνωστή ως PCA από την ονομασία που έδωσε ο Harold Hotelling αργότερα αναπτύσσοντας ξεχωριστά την ίδια μεθοδολογία [60]. Πιο συγκεκριμένα για να κατανοήσουμε τον τρόπο λειτουργίας της μεθόδου θα πρέπει να διευκρινίσουμε κάποιους όρους.

Διακύμανση (Variance) ή διασπορά είναι το μέτρο που μας δείχνει τη διαφορετικότητα των δεδομένων, δηλαδή πόσο απλωμένα είναι στο σύνολό τους από τη μέση τιμή και ισούται με

$$Var(x) = \frac{\sum_{i=0}^n (x_i - \bar{x})^2}{n}$$

όπου x είναι το σύνολο των δεδομένων μας, x_i είναι η τιμή του δείγματος στην i -θέση, \bar{x} είναι η μέση τιμή και n το πλήθος των δειγμάτων στο σύνολο.

Τυπική απόκλιση (Standard Deviation) είναι η τετραγωνική ρίζα της διακύμανσης και δείχνει το ποσό της μεταβολής ή τη διασπορά των δεδομένων.

$$SD = \sqrt{Var}$$

Συνδιακύμανση (Covariance) είναι ένα μέτρο που μας δείχνει πόσο συσχετισμένες είναι οι παρατηρήσεις δυο διαφορετικών συνόλων δεδομένων x, y και δίνεται από τον τύπο

$$Cov(x, y) = \frac{\sum_{i=0}^n (x_i - \bar{x})(y_i - \bar{y})}{n}$$

όπου x, y είναι οι αντίστοιχες τιμές των δειγμάτων στις συγκεκριμένες θέσεις.

Ο σκοπός λοιπόν της μεθόδου είναι να βρει καινούριες συνιστώσες στα δεδομένα έτσι ώστε αυτές να είναι ορθογώνιες, άρα και γραμμικώς ανεξάρτητες, και να τις ταξινομήσει βάσει της διακύμανσης τους. Οπότε η πρώτη συνιστώσα να περιεχέει την περισσότερη πληροφορία του αρχικού μας συνόλου παρατηρήσεων.

Πως δουλεύει η Ανάλυση Κύριων Συνιστωσών

1. Κανονικοποιούμε τα δεδομένα μας.
2. Βρίσκουμε το μητρώο συνδιακύμανσης των δεδομένων μας.
3. Βρίσκουμε τα ιδιοδιανύσματα του μητρώου και τις αντίστοιχες ιδιοτιμές τους.
4. Ταξινομούμε τα ιδιοδιανύσματα σε φθίνουσα σειρά βάσει των ιδιοτιμών τους.
5. Επιλέγουμε τα πρώτα m διανύσματα ανάλογα με το ποσοστό της πληροφορίας που θέλουμε να συγκρατήσουμε.
6. Μετασχηματίζουμε τα αρχικά n -διάστατα δεδομένα σε m -διάστατα ($m < n$).

3.2 Διαδοχική Ανάλυση Κύριων Συνιστωσών χωρίς μητρώο Συνδιακύμανσης

Η Ανάλυση Κύριων Συνιστωσών χρειάζεται πρόσβαση σε όλα τα δεδομένα ώστε να γίνει η ιδιοανάλυση του μητρώου συνδιακύμανσης. Εξαιτίας του γεγονότος ότι δεν προϋπάρχουν δεδομένα κίνησης από τον αισθητήρα επιτάχυνσης, έπρεπε να βρούμε ένα αλγόριθμο που να μπορεί να εκτιμήσει τις κύριες συνιστώσες βάσει των παρατηρήσεων που λαμβάνουμε διαδοχικά. Στην προκειμένη περίπτωση χρησιμοποιήσαμε την Διαδοχική Ανάλυση Κύριων Συνιστωσών χωρίς μητρώο Συνδιακύμανσης (Candid Covariance-free Incremental Principal Component Analysis CCIPCA) [61] για την εκτίμηση της κύριας συνιστώσας.

Αν και υπάρχουν αρκετές διαδοχικές μέθοδοι που είτε χρησιμοποιούν είτε όχι μητρώο συνδιακύμανσης [62, 63, 64, 65], ο CCIPCA συγκλίνει γρηγορότερα [66] όποτε και προτιμήθηκε για αυτή την εφαρμογή.

Έχουμε διαδοχικά διανύσματα k -διάστατα και ταυτόχρονα διαδοχικά αφαιρούμε τη μέση τιμή. Έχουμε λοιπόν A το μητρώο συνδιακύμανσης. Γνωρίζουμε πως για ένα ιδιοδιάνυσμα x του μητρώου A ισχύει

$$\lambda x = Ax$$

όπου λ η αντίστοιχη ιδιοτιμή. Αντικαθιστώντας το A με τον μητρώο συνδιακύμανσης και το x με την εκτίμηση του στη χρονική στιγμή i έχουμε

$$v(n) = \frac{1}{n} \sum_{i=0}^n u(i)u^T(i)x(i)$$

$v(n)$ είναι η n -οστή εκτίμηση του v και μετά με τους τύπους $\lambda = \|v\|$, $x = v/\|v\|$ βρίσκουμε ιδιοδιάνυσμα και ιδιοτιμή. Βασιζόμενοι στο $x = v/\|v\|$ παίρνουμε $x(i) = v(i-1)/\|v(i-1)\|$ και φτάνουμε στην ισότητα

$$v(n) = \frac{1}{n} \sum_{i=0}^n u(i)u^T(i) \frac{v(i-1)}{\|v(i-1)\|}$$

και αφότου θέσουμε $v(0) = v(1)$ ως την πρώτη διεύθυνση των δεδομένων μας έχουμε σε αναδρομική μορφή

$$v(n) = \frac{n-1}{n}v(n-1) + \frac{1}{n}u(n)u^T(n) \frac{v(n-1)}{\|v(n-1)\|}$$

Τέλος για να βελτιώσουμε τον αλγόριθμο περαιτέρω χρησιμοποιούμε βάρη δίνοντας μικρότερη βαρύτητα στα παλιότερα δεδομένα και μεγαλύτερη σε αυτά που έρχονται σε κάθε βήμα. Αυτό γιατί τα δείγματα μας στον τύπο είναι ισοβαρή κάτι που στην αρχή της εκτίμησης μας δίνει πολύ μακριά αποτελέσματα από την πραγματική τιμή. Τελικά ο τύπος μας διαμορφώνεται ως εξής

$$v(n) = \frac{n-1-l}{n}v(n-1) + \frac{1+l}{n}u(n)u^T(n) \frac{v(n-1)}{\|v(n-1)\|}$$

$\frac{n-1-l}{n}$	$\frac{v(n)}{n}$	Η εκτίμηση της κύριας συνιστώσας στη χρονική στιγμή n
$\frac{1+l}{n}$	l	Τα βάρη αντίστοιχα της προηγούμενης εκτίμησης και των νέων δεδομένων
l	$u(n)$	Μια θετική παράμετρος που δίνει μεγαλύτερο βάρος στα τελευταία δεδομένα
$u(n)$	$(\cdot)^T$	Η n -οστή παρατήρηση που έχει αφαιρεθεί η μέση τιμή
$(\cdot)^T$	$\ \cdot\ $	Ο ανάστροφος τελεστής
$\ \cdot\ $		Ο τελεστής της L2 νόρμας

Πίνακας 3.1: Επεξήγηση όρων

3.3 Σωρευτικό Αθροιστικό Διάγραμμα Ελέγχου

Ο αλγόριθμος Cumulative Sum γνωστός και ως CuSum αναπτύχθηκε από τον E.S. Page [22] και χρησιμοποιείται για τον εντοπισμό της αλλαγής δηλαδή την απόκλιση από μια ορισμένη επιθυμητή τιμή, [67] στη διαδοχή των παρατηρήσεων. Χρησιμοποιείται κατά κόρον σε γραμμές παραγωγής για τον ποιοτικό έλεγχο.

Πιο συγκεκριμένα, το σωρευτικό αθροιστικό διάγραμμα ανιχνεύει τις αποκλίσεις από την τιμή στόχο. Για να χρησιμοποιήσουμε ένα τέτοιο διάγραμμα πρέπει να ξέρουμε τι αλλαγές θέλουμε να μπορούμε να εντοπίσουμε, συνήθως είναι μισή ή μια τυπική απόκλιση. Επίσης πρέπει να θέσουμε τα όρια ελέγχου, κατά γενική πρακτική είναι τρεις ή τέσσερις τυπικές αποκλίσεις, και σε αυτό το όριο θα χρειαστεί παρέμβαση. Αν το δείγμα μας είναι εντός του επιτρεπτού ορίου καλής λειτουργίας τότε αφαιρούμε από το σωρευτικό άθροισμα εκτός και αν είναι μηδέν οπότε και παραμένει ως έχει. Αν το δείγμα είναι εκτός των ορίων αλλαγής τότε προστίθεται μια τιμή. Όταν το άθροισμα περάσει το όριο ελέγχου που έχουμε θέσει τότε μας ειδοποιεί πως έχουμε αποκλίνει αρκετά από το στόχο μας, για να μπορέσουμε να το διορθώσουμε. Υπάρχουν δύο σωρευτικά αθροίσματα κάθε φορά, ένα για τις θετικές αποκλίσεις (πάνω από την ιδανική τιμή) αλλά και για τις αρνητικές (τις αποκλίσεις που συμβαίνουν κάτω από την τιμή στόχο).

Πως δουλεύει ο CuSum

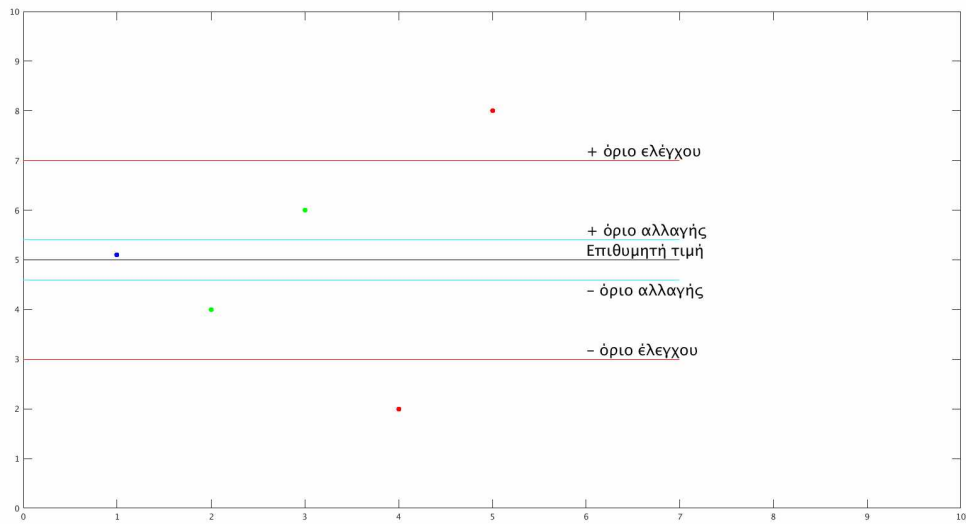
1. Υπολογίζουμε την τυπική απόκλιση
2. Θέτω τα όρια αλλαγής και έλεγχου βάσει της τυπικής απόκλισης
3. Υπολογίζω το θετικό και αρνητικό σωρευτικό άθροισμα με τον εξής τύπο

$$Sp_0, Sn_0 = 0$$

$$Sp_{n+1} = \max(0, Sp_n + x_n - w_n)$$

$$Sn_{n+1} = \min(0, Sn_n + x_n - w_n)$$

4. Ελέγχω αν η τιμή των αθροισμάτων είναι μεγαλύτερη από το θετικό ή αρνητικό όριο ελέγχου και σηματοδοτώ αλλαγή



Εικόνα 3.1: Παράδειγμα λειτουργίας του Σωρευτικού Αθροιστικού Διαγράμματος Ελέγχου

4.1 Γλώσσα Προγραμματισμού

Για τον προγραμματισμό της εφαρμογής αυτής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java, μια γλώσσα γενικής χρήσης βασισμένη στις κλάσεις και την αντικειμενοστρέφεια. Η Java αναπτύχθηκε αρχικά από την Sun Microsystems η οποία αγοράστηκε από την Oracle. Μπορεί να λειτουργήσει σε οποιοδήποτε σύστημα υπάρχει Java Virtual Machine (JVM) κάτι που την καθιστά ιδιαίτερα ελκυστική επιλογή. Η Java είναι και η κατεξοχήν γλώσσα προγραμματισμού για εφαρμογές στην πλατφόρμα του Android με τη μεγαλύτερη υποστήριξη από τον κατασκευαστή της. Χρησιμοποιήσαμε επίσης τη βιβλιοθήκη Efficient Java Matrix Library (EJML)[68]. Τέλος το περιβάλλον ανάπτυξης που χρησιμοποιήθηκε είναι το Android Studio [69] έκδοση 3.5.2 που παρέχεται δωρεάν από την ίδια εταιρία που αναπτύσσει το λειτουργικό σύστημα Android.

4.2 Αρχική Οθόνη Εφαρμογής

Στην εικόνα 4.1 βλέπουμε το γραφικό περιβάλλον της εφαρμογής που έχουμε φτιάξει. Επάνω φαίνεται το όνομα της εφαρμογής “sensors”. Ακριβώς από κάτω βλέπουμε τρία κουμπιά με ονόματα Start και Stop που αντιστοίχως ξεκινούν και σταματούν την καταγραφή των δεδομένων σε αρχεία τύπου .csv στην εξωτερική μνήμη του κινητού τηλεφώνου. Ένα ακόμη κουμπί με ονομασία Initialize... είναι απενεργοποιημένο γιατί δεν έχει ακόμη υλοποιηθεί η λειτουργία του περισσότερο στο κεφάλαιο 6.

Έπειτα βλέπουμε τις τρεις τιμές που μας επιστρέφει ο αισθητήρας γραμμικής επιτάχυνσης και ακριβώς από κάτω έχουμε τις τιμές των παραμέτρων lamda, drift, threshold και το κουμπί που ενημερώνει τις μεταβλητές της εφαρμογής αν τις αλλάξουμε κατά τη διάρκεια λειτουργίας της.

Στο κάτω μέρος της οθόνης φαίνονται δυο πλαίσια εισαγωγής κειμένου με δύο αντίστοιχα κουτιά επιλογής. Το πάνω πλαίσιο περιέχει το νούμερο της επαφής σε περίπτωση ανάγκης και το κάτω τη διεύθυνση ηλεκτρονικού ταχυδρομείου αυτής. Τα κουτιά επιλογής στο πλάι

sensors

START INITIALIZE... STOP

x:-0.028362095
y:-0.010486722
z:-0.04422283

lamda UPDATE

drift

threshold

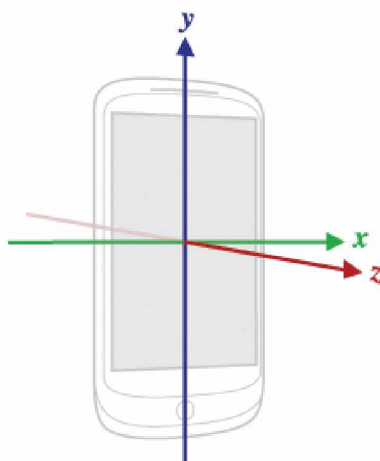
SAVE

Εικόνα 4.1: Κύρια οθόνη της εφαρμογής.

σηματοδοτούν τον τρόπο ή τους τρόπους που θέλουμε να γίνει επικοινωνία σε περίπτωση ανάγκης, είτε μέσω γραπτού μηνύματος είτε μέσω ηλεκτρονικού ταχυδρομείου. Τέλος το κουμπί κάτω δεξιά με το κείμενο Save αποθηκεύει τις προτιμήσεις επικοινωνίας σε μνήμη που ακόμα και αν η εφαρμογή τερματιστεί με όχι κανονικό τρόπο αυτές θα παραμείνουν.

4.3 Συλλογή Δεδομένων

Γίνεται μια παρουσίαση των δεδομένων (εικ 4.1) καθώς λαμβάνονται από τον αισθητήρα και καταγράφονται σε ένα αρχείο τύπου comma separated values (csv) για μετέπειτα έλεγχο σφαλμάτων στην εφαρμογή. Ο αισθητήρας επιτάχυνσης που ενσωματώνει το κινητό τηλέφωνο επιστρέφει κάθε χρονική στιγμή τρεις τιμές που δείχνουν την τιμή της επιτάχυνσης σε κάθε μια από τις τρεις κατευθύνσεις $[x, y, z]$ (εικόνα 4.2).



Εικόνα 4.2: Σύστημα συντεταγμένων (σε σχέση με τη συσκευή) που χρησιμοποιείται από το Sensor API

4.4 Ανάλυση των δεδομένων σε πραγματικό χρόνο

Τα δεδομένα σε κάθε χρονική στιγμή αποθηκεύονται σε ένα τρισδιάστατο διάνυσμα και υπολογίζεται η μέση τιμή τους διαδοχικά. Στο επόμενο βήμα παίρνουμε το κανονικοποιημένο διάνυσμα και υπολογίζουμε την πρώτη κύρια συνιστώσα σύμφωνα με τον επαναληπτικό τύπο του CCIPCA

$$v(n) = \frac{n-1-l}{n}v(n-1) + \frac{1+l}{n}u(n)u^T(n)\frac{v(n-1)}{\|v(n-1)\|}$$

όπου $v(n)$ είναι η εκτίμηση της κύριας συνιστώσας στο n -οστό δείγμα, και

$$\frac{n-1-l}{n}, \frac{1+l}{n}$$

είναι τα βάρη της προηγούμενης εκτίμησης και των νέων δεδομένων αντίστοιχα, l είναι μια θετική παράμετρος που δίνει μεγαλύτερο “βάρος” στις νεότερες τιμές και βοηθάει να συγκλίνει γρηγορότερα ο αλγόριθμος, $u(n)$ το n -οστό διάνυσμα με τις παρατηρήσεις έχοντας αφαιρέσει τη μέση τιμή καθώς και ο ανάστροφος του. Τέλος $\|v(n-1)\|$ η νόρμα της προηγούμενης εκτίμησης. Αφού υπολογιστεί η κύρια συνιστώσα σε κάθε βήμα τότε υπολογίζεται η προβολή καθώς και η μέση τιμή τους.

$$proj(n) = v(n) * u(n)$$

Τα οποία παίρνει ως είσοδο η συνάρτηση του CuSum. Επειδή ο CuSum χρειάζεται ένα εκτιμώμενο στόχο που δεν τον γνωρίζουμε εκ των προτέρων χρησιμοποιούμε κάθε φορά τη νέα μέση τιμή των δειγμάτων που έχουμε συλλέξει ως τη δεδομένη στιγμή.

Υπολογίζουμε την τρέχουσα διακύμανση με την προσθήκη κάθε νέας προβολής

$$currVar = \sigma^2 = \frac{\sum_{i=0}^n (x_i - \bar{x})^2}{n - 1}$$

και έπειτα την τυπική απόκλιση

$$currStd = \sigma = \sqrt{currVar}$$

και δύο ακόμα παραμέτρους που συναρτήσει της τυπικής απόκλισης μας δίνουν τα όρια αλλαγής και ελέγχου, την παράμετρο *drift* που μας δίνει το όριο αλλαγής $slack = drift * currStd$ και το όριο ελέγχου (*threshold*) που το ορίζουμε ως $threshold * currStd$. Υπολογίζουμε τα δύο αθροίσματα για τις θετικές τιμές αλλά και τις αρνητικές

$$Sp(n) = \max(0, Sp(n-1) + proj(n) - projMean - slack), Sp(0) = 0$$

$$Sn(n) = \min(0, Sn(n-1) + proj(n) - projMean + slack), Sn(0) = 0$$

Τέλος ελέγχουμε αν κατά απόλυτη τιμή τα αθροίσματα είναι μεγαλύτερα από το όριο ελέγχου που έχουμε θέσει και σε αυτή την περίπτωση εμφανίζεται ένα μήνυμα στο αρχείο καταγραφής (ή οποιαδήποτε άλλη ενέργεια θέλουμε να υλοποιήσουμε).

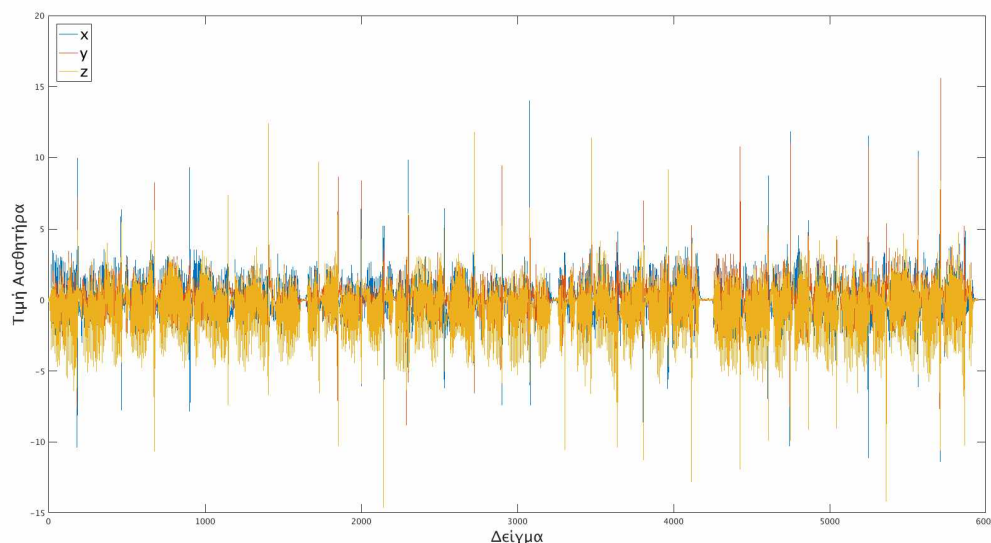
Αποτελέσματα και Συμπεράσματα

5.1 Υλικό

Η κινητή συσκευή στην οποία ελέγχθηκε ο κώδικας της εφαρμογής και η λειτουργία του είναι το OnePlus One model A0001 με την έκδοση 9 “Pie” του λειτουργικού συστήματος Android.

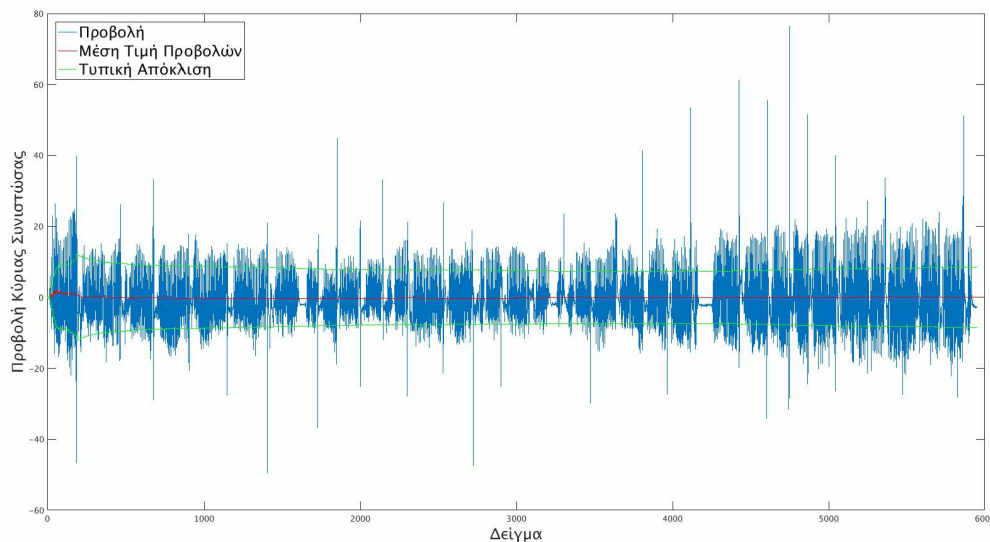
5.2 Ανάλυση των αποτελεσμάτων

Για να κρίνουμε την αποτελεσματικότητα της εφαρμογής, συλλέξαμε ένα πλήθος δεδομένων και τα εξετάσαμε ως προς την εγκυρότητα του αποτελέσματος του αλγορίθμου.



Εικόνα 5.1: Οι τιμές του αισθητήρα επιτάχυνσης στους άξονες x, y, z .

Στο παρακάτω διάγραμμα 5.2 φαίνονται οι προβολές που μας επιστρέφει ο CCIPCA καθώς και η μέση τιμή τους και η τυπική απόκλιση. Κρίναμε βάση των διαγραμμάτων όπως

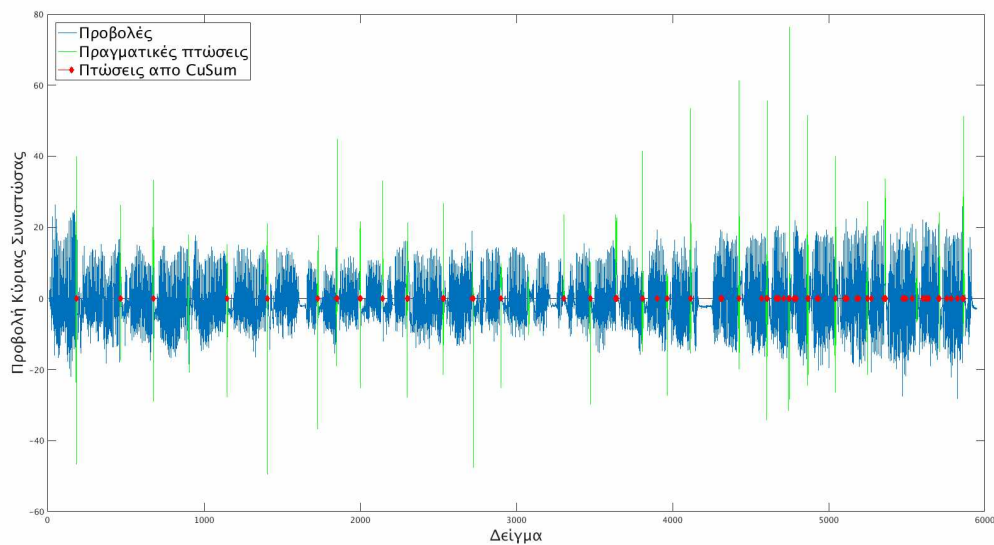


Εικόνα 5.2: Προβολές κύριας συνιστώσας, μέση τιμή και τυπική απόκλιση.

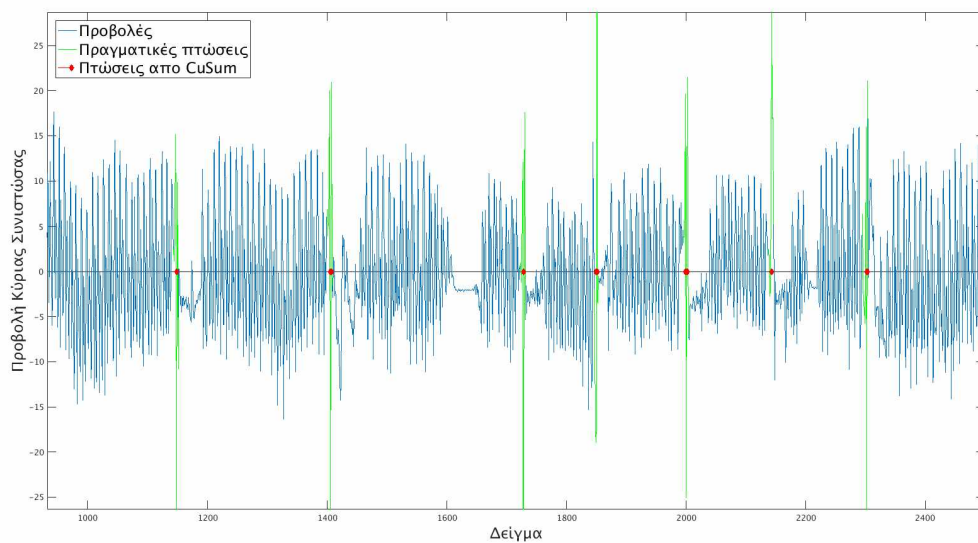
φαίνεται στην εικόνα 5.1 των διανυσμάτων επιτάχυνσης καθώς και από βίντεο τις στιγμές που υπήρχαν πτώσεις και δημιουργήσαμε ένα διάνυσμα που έχει την τιμή 1 στα δείγματα που είχαν πτώση και -1 στα υπόλοιπα. Επίσης δημιουργήσαμε ένα διάνυσμα με μηδενικά σε κάθε θέση πέρα από τις θέσεις που ικανοποιείται ο CuSum που δίναμε την τιμή 1. Βρίσκοντας το εσωτερικό γινόμενο αυτών των δύο διανυσμάτων δημιουργήσαμε ένα μέτρο που αυξάνεται κατά ένα για κάθε πτώση που αναγνωρίστηκε και μειώνεται κατά ένα για κάθε πτώση που βρέθηκε ενώ δεν υπήρξε. Με αυτό τον τρόπο τρέξαμε επαναληπτικά για ένα εύρος τιμών των μεταβλητών όριο αλλαγής και όριο ελέγχου (drift, threshold) έτσι ώστε να βρούμε τις βέλτιστες τιμές για τη μεγαλύτερη ακρίβεια.

Οι τιμές που βρήκαμε είναι $drift = 2$ και $threshold = 0.48$. Με αυτές τις τιμές η εφαρμογή μας έχει ευστοχία στην πρόβλεψη των πτώσεων της τάξης του 95% όπως φαίνεται στο διάγραμμα 5.3. Με μπλέ όπως παραπάνω είναι η τιμή της προβολής στην νέα κύρια συνιστώσα με πράσινο έχουμε χαρακτηρίσει εμείς με το χέρι τα σημεία με πραγματική πτώση και με κόκκινο βλέπουμε τα σημεία που βρήκε πτώση το Σωρευτικό Αθροιστικό Διάγραμμα Ελέγχου. Στο διάγραμμα 5.4 μπορούμε να διακρίνουμε σε μεγέθυνση τι συμβαίνει στο παράθυρο δειγμάτων από 1000 έως 2500.

Για να μετρήσουμε την ακρίβεια του αλγορίθμου μας δημιουργήσαμε τέσσερις μετρητές, True Positive, False Positive, True Negative και False Negative. Ανάλογα με το τελικό αποτέλεσμα του CuSum σε κάθε βήμα αυξάνουμε τον ανάλογο μετρητή κατά μία μονάδα. Συγκεκριμένα ο True Positive αυξάνεται αν υπάρχει πτώση και την βρήκε η εφαρμογή μας. Ο False Positive αυξάνεται όταν ο αλγόριθμος διαπιστώσει πτώση ενώ δεν υπήρχε. True Negative όταν δεν υπήρχε πτώση και ορθά δεν έδειξε πτώση ο αλγόριθμος και τέλος False Negative όταν ενώ υπήρχε πτώση ο CuSum έδειξε πως δεν υπήρχε.



Εικόνα 5.3: Τα σημεία που έδειξε ο αλγόριθμος πτώση.



Εικόνα 5.4: Μεγέθυνση στο δείγματα 1000-2500.

Από το παραπάνω γράφημα 5.3 παίρνουμε τον πίνακα:

	Πτώση	Όχι Πτώση
CuSum θετικό	28	15
CuSum αρνητικό	3	5836

$$\text{Ευαισθησία} = \text{Sensitivity} = \frac{TP}{P} = \frac{28}{31} \approx 0,903$$

$$\text{Ειδικότητα} = \text{Specificity} = \frac{TN}{N} = \frac{5836}{5851} \approx 0,997$$

$$\text{Ακρίβεια} = \text{Accuracy} = \frac{TP + TN}{P + N} = \frac{5864}{5882} \approx 0,997$$

επειδή όμως τα σετ των δεδομένων μας είναι αρκετά δυσανάλογα στο πλήθος (πτώσεις, όχι πτώσεις) θα χρησιμοποιήσουμε τον τύπο της ισορροπημένης ακρίβειας

$$\text{Ισορροπημένη Ακρίβεια} = \text{Balanced Accuracy} = \frac{\frac{TP}{P} + \frac{TN}{N}}{2} = \frac{\frac{28}{31} + \frac{5836}{5851}}{2} \approx 0,95$$

Όπως βλέπουμε ο αλγόριθμος βρίσκει τις πτώσεις και μή πτώσεις με ακρίβεια της τάξης του 95%.

5.3 Κατανάλωση μπαταρίας

Ένα μείζον πρόβλημα σε οποιαδήποτε εφαρμογή είναι η κατανάλωση της μπαταρίας. Ειδικά ένα κινητό τηλέφωνο θα θέλαμε, ιδανικά, να μπορεί να χρησιμοποιείται για μία ολόκληρη ημέρα χωρίς φόρτιση. Ένα θέμα που πρέπει να εξεταστεί λοιπόν στην εφαρμογή είναι η κατανάλωση της μπαταρίας, κάτι που αποδεικνύεται εξαιρετικά δύσκολο γιατί την κατανάλωση την επηρεάζουν πάρα πολύ παράγοντες και δεν γίνεται να απομονώσουμε μόνο τη δική μας εφαρμογή. Για να έχουμε όμως μια γενική ματιά στην κατανάλωση χρησιμοποιήσαμε δύο εφαρμογές που μετρούν την κατανάλωση της μπαταρίας στο τηλέφωνο AccuBattery, GSam Battery Monitor και την λειτουργικότητα αυτή του ίδιου του λειτουργικού. Για περίπου είκοσιένα λεπτά χρήσης της εφαρμογής βρέθηκαν τα παρακάτω.

Πρόγραμμα	Ποσοστό	Κατανάλωση σε mA
Android Battery Monitor	1%	28mA
AccuBattery	1.4%	40mA
GSam Battery Monitor	0,7%	19,6mA

Πίνακας 5.1: Κατανάλωση μπαταρίας

Φαίνεται από τον πίνακα 5.1 ότι η κατανάλωση κυμαίνεται περίπου στα 83mAh. Αν αναλογιστούμε ότι η μέση χωρητικότητα μιας μπαταρίας είναι τα 3000mAh τότε θα μπορούσε να λειτουργεί η εφαρμογή για περίπου 36 ώρες. Σίγουρα αυτή η μέθοδος μέτρησης δεν είναι

η καλύτερη αλλά είναι μια πρώτη μάτια. Η κατανάλωση είναι κάτι που θα πρέπει να μελετηθεί περισσότερο και να βελτιστοποιηθεί ο κώδικας για να ελαχιστοποιηθεί.

Συζήτηση και μελλοντική μελέτη

Όπως είδαμε, η εφαρμογή ανταποκρίνεται στις απαιτήσεις της συγκεκριμένης πτυχιακής εργασίας με μεγάλη ακρίβεια χωρίς να έχει μεγάλη κατανάλωση ισχύος της κεντρικής μονάδας επεξεργασίας του κινητού τηλεφώνου, γεγονός που βοηθάει στην μικρότερη κατανάλωση της μπαταρίας. Ωστόσο ο αλγόριθμος, μπορεί να βελτιστοποιηθεί περαιτέρω έτσι ώστε να έχει πιθανώς και εμπορική χρήση. Ήδη από τη συγγραφή της εφαρμογής και της διπλωματικής, παρουσιάστηκαν κάποιες πιθανές βελτιώσεις που θα πρέπει να τις επισκεφτούμε στο μέλλον. Μερικές από αυτές είναι οι εξής:

- Εξέταση διαφορετικών αλγορίθμων μετά από έρευνα στη βιβλιογραφία για να βρεθεί κάποιος που να συγκλίνει γρηγορότερα ή κάποιος πιο κατάλληλος για τη εφαρμογή μας πχ. ο [70] που φαίνεται πως έχει καλύτερα αποτελέσματα από τον CCIPCA.
- Να γίνει εκτενέστερος έλεγχος της χρήσης της μπαταρίας της συσκευής από την εφαρμογή και να μειωθεί η κατανάλωση.
- Στην έκδοση 9 “Pie” του λειτουργικού συστήματος Android έχουν γίνει κάποιες αλλαγές σχετικά με τη χρήση των πόρων του τηλεφώνου από εφαρμογές που είναι στο παρασκήνιο, οπότε πρέπει να προγραμματίσουμε την εφαρμογή να τρέχει στο προσκήνιο “foreground service” ώστε να μην χάνει τη συνεχή ροή των δεδομένων από τον αισθητήρα της επιτάχυνσης.
- Για τον ίδιο σκοπό, θετικό θα ήταν να κοιτάξουμε πως επηρεάζεται η ακρίβεια της εφαρμογής με λιγότερες μετρήσεις από τον αισθητήρα στον ίδιο χρόνο.
- Επειδή ήδη κάνουμε χρήση αλγορίθμου μείωσης των μεταβλητών, θα μπορούσαμε στο μέλλον να ενσωματώσουμε και άλλους αισθητήρες. Είτε από τους ήδη υπάρχοντες στο τηλέφωνο όπως γυροσκόπιο, ή τον αισθητήρα εγγύτητας είτε εξωτερικούς αισθητήρες από “έξυπνα” ρολόγια ή “φορετούς” αισθητήρες στα ρούχα κλπ, με σκοπό να βελτιώσουμε την ποιότητα της εφαρμογής ή να την επεκτείνουμε.

- Επιπλέον, πιθανή χρήση θα ήταν και η συλλογή δεδομένων για άλλες έρευνες ακόμα και σε διαφορετικά πεδία και σε μεγαλύτερη κλίμακα, σεβόμενοι πάντα την ιδιωτικότητα των ατόμων.

Αυτές είναι μερικές βελτιώσεις και μελλοντικές ιδέες που θα πρέπει να μας κινήσουν το ενδιαφέρον και να προσπαθήσουμε να τις υλοποιήσουμε.

Βιβλιογραφία

- [1] R. Ling and J. Donner, *Mobile Communication*. John Wiley & Sons, May 2013.
- [2] J. E. Katz, “Mobile Media and Communication: Some Important Questions,” *Communication Monographs*, vol. 74, pp. 389–394, Sept. 2007.
- [3] S. Etzo and G. Collender, “The mobile phone ‘revolution’ in Africa: Rhetoric or reality?,” *African Affairs*, vol. 109, pp. 659–668, Oct. 2010.
- [4] R. W. Sattin, “Falls among older persons: A public health perspective,” *Annual Review of Public Health*, vol. 13, pp. 489–508, 1992.
- [5] J. M. Hausdorff, D. A. Rios, and H. K. Edelberg, “Gait variability and fall risk in community-living older adults: A 1-year prospective study,” *Archives of Physical Medicine and Rehabilitation*, vol. 82, pp. 1050–1056, Aug. 2001.
- [6] S. K. Inouye, C. J. Brown, and M. E. Tinetti, “Medicare Nonpayment, Hospital Falls, and Unintended Consequences,” *New England Journal of Medicine*, vol. 360, pp. 2390–2393, June 2009.
- [7] Y. Dionyssiotis, “Analyzing the problem of falls among older people,” *International Journal of General Medicine*, vol. 5, pp. 805–813, 2012.
- [8] R. Boyd and J. A. Stevens, “Falls and fear of falling: Burden, beliefs and behaviours,” *Age and Ageing*, vol. 38, pp. 423–428, July 2009.
- [9] J. E. Dunn, M. A. Rudberg, S. E. Furner, and C. K. Cassel, “Mortality, disability, and falls in older persons: The role of underlying disease and disability,” *American Journal of Public Health*, vol. 82, pp. 395–400, Mar. 1992.
- [10] S. V. Georgakopoulos, S. K. Tasoulis, I. Maglogiannis, and V. P. Plagianakos, “On-Line Fall Detection via Mobile Accelerometer Data,” in *Artificial Intelligence Applications and Innovations* (R. Chbeir, Y. Manolopoulos, I. Maglogiannis, and R. Al-

-
- hajj, eds.), IFIP Advances in Information and Communication Technology, (Cham), pp. 103–112, Springer International Publishing, 2015.
- [11] F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, “Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls,” *PLOS ONE*, vol. 7, p. e37062, May 2012.
- [12] U. Lindemann, A. Hock, M. Stuber, W. Keck, and C. Becker, “Evaluation of a fall detector based on accelerometers: A pilot study,” *Medical and Biological Engineering and Computing*, vol. 43, pp. 548–551, Oct. 2005.
- [13] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, “Wearable Sensors for Reliable Fall Detection,” in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 3551–3554, Jan. 2005.
- [14] A. K. Bourke, J. V. O’Brien, and G. M. Lyons, “Evaluation of a threshold-based triaxial accelerometer fall detection algorithm,” *Gait & Posture*, vol. 26, pp. 194–199, July 2007.
- [15] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, “Comparison of low-complexity fall detection algorithms for body attached accelerometers,” *Gait & Posture*, vol. 28, pp. 285–291, Aug. 2008.
- [16] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, “Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information,” in *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 138–143, June 2009.
- [17] O. Almeida, M. Zhang, and J.-C. Liu, “Dynamic Fall Detection and Pace Measurement in Walking Sticks,” in *2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)*, pp. 204–206, June 2007.
- [18] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, “Barometric pressure and triaxial accelerometry-based falls event detection,” *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 18, pp. 619–627, Dec. 2010.
- [19] M. Á. Estudillo-Valderrama, L. M. Roa, J. Reina-Tosina, and D. Naranjo-Hernández, “Design and Implementation of a Distributed Fall Detection System—Personal Server,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, pp. 874–881, Nov. 2009.

- [20] J. Kang, T. Yoo, and H. Kim, "A Wrist-Worn Integrated Health Monitoring Instrument with a Tele-Reporting Device for Telemedicine and Telecare," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, pp. 1655–1661, Oct. 2006.
- [21] C.-S. Lin, H. C. Hsu, Y.-L. Lay, C.-C. Chiu, and C.-S. Chao, "Wearable device for real-time monitoring of human falls," *Measurement*, vol. 40, pp. 831–840, Nov. 2007.
- [22] S. Luo and Q. Hu, "A dynamic motion pattern analysis approach to fall detection," in *IEEE International Workshop on Biomedical Circuits and Systems, 2004.*, pp. 1–5, Dec. 2004.
- [23] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement*, vol. 25, pp. R1–20, Apr. 2004.
- [24] M. R. Narayanan, S. R. Lord, M. M. Budge, B. G. Celler, and N. H. Lovell, "Falls management: Detection and prevention, using a waist-mounted triaxial accelerometer," *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2007, pp. 4037–4040, 2007.
- [25] N. Noury, P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau, "A smart sensor based on rules and its evaluation in daily routines," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 4, pp. 3286–3289 Vol.4, Sept. 2003.
- [26] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, and O. Tanaka, "A wearable airbag to prevent fall injuries," *IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 13, pp. 910–914, Nov. 2009.
- [27] C.-C. Wang, C.-Y. Chiang, P.-Y. Lin, Y.-C. Chou, I.-T. Kuo, C.-N. Huang, and C.-T. Chan, "Development of a Fall Detecting System for the Elderly Residents," in *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1359–1362, May 2008.
- [28] G. Williams, K. Doughty, K. Cameron, and D. Bradley, "A smart fall and activity monitor for telecare applications," in *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286)*, vol. 3, pp. 1151–1154 vol.3, Nov. 1998.
- [29] G. Wu, "Distinguishing fall activities from normal activities by velocity characteristics," *Journal of Biomechanics*, vol. 33, pp. 1497–1500, Nov. 2000.

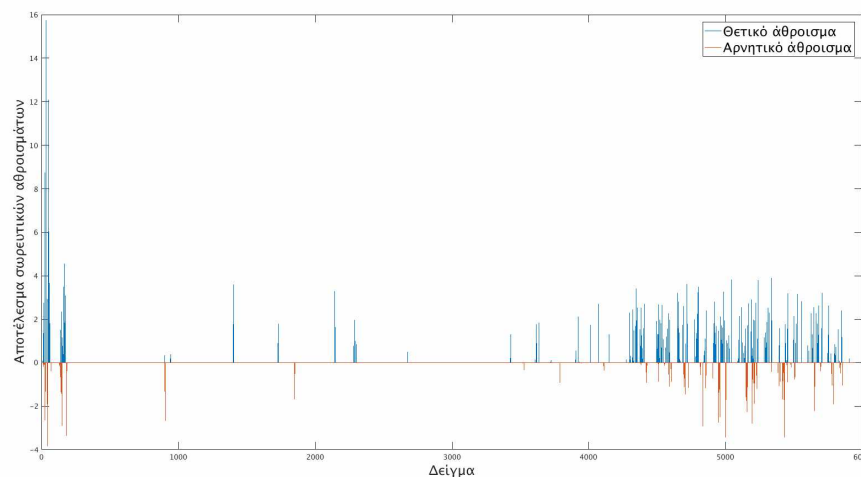
-
- [30] C. Doukas, I. Maglogiannis, P. Tragas, D. Liapis, and G. Yovanof, "Patient Fall Detection using Support Vector Machines," in *Artificial Intelligence and Innovations 2007: From Theory to Applications* (C. Boukis, A. Pnevmatikakis, and L. Polymenakos, eds.), IFIP The International Federation for Information Processing, (Boston, MA), pp. 147–156, Springer US, 2007.
- [31] T. Zhang, J. Wang, L. Xu, and P. Liu, "Fall Detection by Wearable Sensor and One-Class SVM Algorithm," in *Intelligent Computing in Signal Processing and Pattern Recognition: International Conference on Intelligent Computing, ICIC 2006 Kunming, China, August 16–19, 2006* (D.-S. Huang, K. Li, and G. W. Irwin, eds.), Lecture Notes in Control and Information Sciences, pp. 858–863, Berlin, Heidelberg: Springer, 2006.
- [32] T. Lee and A. Mihailidis, "An intelligent emergency response system: Preliminary development and testing of automated fall detection," *Journal of Telemedicine and Telecare*, vol. 11, no. 4, pp. 194–198, 2005.
- [33] S.-G. Miaou, Pei-Hsu Sung, and Chia-Yuan Huang, "A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information," in *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, 2006. D2H2.*, pp. 39–42, Apr. 2006.
- [34] A. Williams, D. Ganesan, and A. Hanson, "Aging in Place: Fall Detection and Localization in a Distributed Smart Camera Network," in *Proceedings of the 15th ACM International Conference on Multimedia, MM '07*, (Augsburg, Germany), pp. 892–901, ACM, 2007.
- [35] C.-W. Lin and Z.-H. Ling, "Automatic Fall Incident Detection in Compressed Video for Intelligent Homecare," in *2007 16th International Conference on Computer Communications and Networks*, pp. 1172–1177, Aug. 2007.
- [36] C.-W. Lin, Z.-H. Ling, Y.-C. Chang, and C. J. Kuo, "Compressed-domain Fall Incident Detection for Intelligent Homecare," *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 49, pp. 393–408, Dec. 2007.
- [37] Huan-Wen Tzeng, Mei-Yung Chen, and J.-Y. Chen, "Design of fall detection system with floor pressure and infrared image," in *2010 International Conference on System Science and Engineering*, pp. 131–135, July 2010.
- [38] G. Diraco, A. Leone, and P. Siciliano, "An active vision system for fall detection and posture recognition in elderly healthcare," in *2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010)*, pp. 1536–1541, Mar. 2010.

- [39] G. Mastorakis and D. Makris, “Fall detection system using Kinect’s infrared sensor,” *Journal of Real-Time Image Processing*, vol. 9, pp. 635–646, Dec. 2014.
- [40] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Monocular 3D Head Tracking to Detect Falls of Elderly People,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6384–6387, Aug. 2006.
- [41] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Fall Detection from Human Shape and Motion History Using Video Surveillance,” in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW’07)*, vol. 2, pp. 875–880, May 2007.
- [42] V. Vishwakarma, C. Mandal, and S. Sural, “Automatic Detection of Human Fall in Video,” in *Pattern Recognition and Machine Intelligence* (A. Ghosh, R. K. De, and S. K. Pal, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 616–623, Springer, 2007.
- [43] Z. Fu, T. Delbruck, P. Lichtsteiner, and E. Culurciello, “An Address-Event Fall Detector for Assisted Living Applications,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, pp. 88–96, June 2008.
- [44] K. K. Delibasis and I. Maglogiannis, “A fall detection algorithm for indoor video sequences captured by fish-eye camera,” in *Proceedings of the 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, BIBE ’15, (USA), pp. 1–5, IEEE Computer Society, Nov. 2015.
- [45] H. Nait-Charif and S. McKenna, “Activity summarisation and fall detection in a supportive home environment,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 4, pp. 323–326 Vol.4, Aug. 2004.
- [46] B. U. Töreyn, Y. Dedeoğlu, and A. E. Çetin, “HMM Based Falling Person Detection Using Both Audio and Video,” in *Computer Vision in Human-Computer Interaction* (N. Sebe, M. Lew, and T. S. Huang, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 211–220, Springer, 2005.
- [47] D. Anderson, J. M. Keller, M. Skubic, X. Chen, and Z. He, “Recognizing Falls from Silhouettes,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6388–6391, Aug. 2006.
- [48] N. Thome and S. Miguet, “A HHMM-Based Approach for Robust Fall Detection,” in *Robotics and Vision 2006 9th International Conference on Control, Automation*, pp. 1–8, Dec. 2006.

-
- [49] R. Cucchiara, A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation," *Expert Systems*, vol. 24, no. 5, pp. 334–345, 2007.
- [50] B. Jansen and R. Deklerck, "Context aware inactivity recognition for visual fall detection," in *2006 Pervasive Health Conference and Workshops*, pp. 1–4, Nov. 2006.
- [51] Y.-t. Hsu, J.-w. Hsieh, H.-f. Kao, and H.-y. M. Liao, "Human Behavior Analysis Using Deformable Triangulations," in *2005 IEEE 7th Workshop on Multimedia Signal Processing*, pp. 1–4, Oct. 2005.
- [52] L. Hazelhoff, J. Han, and P. H. N. de With, "Video-Based Fall Detection in the Home Using Principal Component Analysis," in *Advanced Concepts for Intelligent Vision Systems* (J. Blanc-Talon, S. Bourennane, W. Philips, D. Popescu, and P. Scheunders, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 298–309, Springer, 2008.
- [53] C.-L. Liu, C.-H. Lee, and P.-M. Lin, "A fall detection system using k-nearest neighbor classifier," *Expert Systems with Applications*, vol. 37, pp. 7174–7181, Oct. 2010.
- [54] C. Zhang, Y. Tian, and E. Capezuti, "Privacy Preserving Automatic Fall Detection for Elderly Using RGBD Cameras," in *Computers Helping People with Special Needs* (K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 625–633, Springer, 2012.
- [55] M. Alwan, P. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A Smart and Passive Floor-Vibration Based Fall Detector for Elderly," in *2006 2nd International Conference on Information Communication Technologies*, vol. 1, pp. 1003–1007, Apr. 2006.
- [56] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive Computing*, vol. 3, pp. 42–47, Apr. 2004.
- [57] A. Sixsmith, N. Johnson, and R. Whatmore, "Pyroelectric IR sensor arrays for fall detection in the older population," *Journal De Physique Iv*, vol. 128, pp. 153–160, Sept. 2005.
- [58] Y. Li, K. C. Ho, and M. Popescu, "A Microphone Array System for Automatic Fall Detection," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 1291–1301, May 2012.
- [59] K. P. F.R.S, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, pp. 559–572, Nov. 1901.

- [60] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.
- [61] J. Weng, Y. Zhang, and W.-S. Hwang, “A Fast Algorithm for Incremental Principal Component Analysis,” in *Intelligent Data Engineering and Automated Learning* (J. Liu, Y.-m. Cheung, and H. Yin, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 876–881, Springer, 2003.
- [62] E. Oja, *Subspace Methods of Pattern Recognition*. Research Studies Press, Dec. 1983.
- [63] T. P. Krasulina, “The method of stochastic approximation for the determination of the least eigenvalue of a symmetrical matrix,” *USSR Computational Mathematics and Mathematical Physics*, vol. 9, pp. 189–195, Jan. 1969.
- [64] Haitao Zhao, Pong Chi Yuen, and J. Kwok, “A novel incremental principal component analysis and its application for face recognition,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 36, pp. 873–886, Aug. 2006.
- [65] W. Li, C. Shuo, and W. Cheng-dong, “An Accurate Incremental Principal Component Analysis method with capacity of update and downdate,”
- [66] Y. Zhang and J. Weng, “Convergence analysis of complementary candid incremental principal component Analysis,” *Computing in Science and Engineering - C in S&E*, Jan. 2001.
- [67] O. A. Grigg, V. T. Farewell, and D. J. Spiegelhalter, “Use of risk-adjusted CUSUM and RSPRTcharts for monitoring in medical contexts,” *Statistical Methods in Medical Research*, vol. 12, pp. 147–170, Apr. 2003.
- [68] P. Abeles, *Efficient Java Matrix Library*. Abeles, Peter, 2019.
- [69] “Documentation — Android Developers.” <https://developer.android.com/docs>.
- [70] S. Yan and X. Tang, “Largest-eigenvalue-theory for incremental principal component analysis,” in *IEEE International Conference on Image Processing 2005*, vol. 1, pp. I–1181, Sept. 2005.

8.1 Συμπληρωματικά Διαγράμματα



Εικόνα 8.1: Το θετικό και αρνητικό σωρευτικό άθροισμα.

8.2 Κώδικα εφαρμογής

```
1 package com.example.android.sensors;  
2  
3 import android.Manifest;  
4 import android.content.Context;  
5 import android.content.Intent;  
6 import android.content.SharedPreferences;  
7 import android.content.pm.PackageManager;  
8 import android.hardware.Sensor;  
9 import android.hardware.SensorEvent;  
10 import android.hardware.SensorEventListener;
```

```
11 import android.hardware.SensorManager;
12 import android.location.Location;
13 import android.location.LocationManager;
14 import android.os.Bundle;
15 import android.os.Environment;
16 import android.os.Looper;
17 import android.provider.Settings;
18 import android.telephony.SmsManager;
19 import android.util.Log;
20 import android.view.View;
21 import android.widget.CheckBox;
22 import android.widget.EditText;
23 import android.widget.TextView;
24 import android.widget.Toast;
25
26 import com.google.android.gms.location.FusedLocationProviderClient;
27 import com.google.android.gms.location.LocationCallback;
28 import com.google.android.gms.location.LocationRequest;
29 import com.google.android.gms.location.LocationResult;
30 import com.google.android.gms.location.LocationServices;
31 import com.google.android.gms.tasks.OnCompleteListener;
32 import com.google.android.gms.tasks.Task;
33
34 import org.ejml.simple.SimpleMatrix;
35
36 import java.io.File;
37 import java.io.FileOutputStream;
38 import java.io.IOException;
39 import java.text.SimpleDateFormat;
40 import java.util.Calendar;
41
42 import androidx.annotation.NonNull;
43 import androidx.appcompat.app.AppCompatActivity;
44 import androidx.core.app.ActivityCompat;
45
46
47 public class MainActivity extends AppCompatActivity implements
    SensorEventListener {
48
49     private static final int PERMISSION_REQUEST_CODE = 100;
50     private SensorManager SM;
51     private Sensor myAccelerometer;
52     private TextView xText, yText, zText;
53     //me threshold 4 kai dirift 1 deixnei na doulevei sxetika kala
54     private double lamda = 3.0, threshold = 0.48, drift = 2;
55     private EditText lamdaText, thresholdText, driftText;
56     private EditText contactNumber, contactEmail;
```

```

57     private CheckBox checkBoxEmail, checkBoxSMS;
58     String filename, fileptoseis;
59     boolean logging = false;
60     SimpleMatrix ipcavOld = new SimpleMatrix(1,3, true, new double[]
        {1,1,1});
61     SimpleMatrix ipcav = new SimpleMatrix(1,3);
62     private double meanOld[] = new double[3], mean[] = new double[3], pca
        [] = new double[3], dian[] = new double[3];
63     private double provoli;
64     private double provoliMean = 0.0;
65     private double currMean = 0.0;
66     private double currVariance = 0.0;
67     SimpleMatrix dataMeanSubtracted = new SimpleMatrix(1,3);
68
69     private double gPosOld =0.0;
70     private double gPos = 0.0;
71     private double gNeg = 0.0;
72     private double gNegOld = 0.0;
73     private double nsamples;
74     private static final String TAG = "MyActivity";
75     private boolean fall = false;
76     FileOutputStream os, fos;
77     //     Vibrator v;
78     FusedLocationProviderClient mFusedLocationClient;
79     SharedPreferences sharedPref;
80     private String conEmail, conNumber;
81     private Location location;
82     private boolean checkSMS, checkEmail;
83
84
85
86     @Override
87     protected void onCreate(Bundle savedInstanceState) {
88         super.onCreate(savedInstanceState);
89         setContentView(R.layout.activity_main);
90
91         SM = (SensorManager) getSystemService(SENSOR_SERVICE);
92         myAccelerometer = SM.getDefaultSensor(Sensor.
            TYPE_LINEAR_ACCELERATION);
93         SM.registerListener(this, myAccelerometer, 150000);
94
95         sharedPref = this.getSharedPreferences(Context.MODE_PRIVATE);
96         conEmail = sharedPref.getString("conEmail", "-");
97         conNumber = sharedPref.getString("conNumber", "-");
98         checkSMS = sharedPref.getBoolean("checkSMS", false);
99         checkEmail = sharedPref.getBoolean("checkEmail", false);
100        checkBoxEmail = findViewById(R.id.checkBoxEmail);

```

```

101     checkBoxSMS = findViewById(R.id.checkBoxSMS);
102     checkBoxEmail.setChecked(checkEmail);
103     checkBoxSMS.setChecked(checkSMS);
104     contactNumber = (EditText) findViewById(R.id.editTextNumber);
105     contactEmail = (EditText) findViewById(R.id.editTextEmail);
106     contactEmail.setText(conEmail);
107     contactNumber.setText(conNumber);
108
109     //         v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
110
111     xText = (TextView) findViewById(R.id.textViewX);
112     yText = (TextView) findViewById(R.id.textViewY);
113     zText = (TextView) findViewById(R.id.textViewZ);
114     lamdaText = (EditText) findViewById(R.id.editText);
115     thresholdText = (EditText) findViewById(R.id.editText2);
116     driftText = (EditText) findViewById(R.id.editText3);
117     lamdaText.setText(String.valueOf(lamda));
118     thresholdText.setText(String.valueOf(threshold));
119     driftText.setText(String.valueOf(drift));
120     nsamples = 0.0;
121     logging = true;
122     filename = "sensors" + getDateTIme();
123     fileptoseis = "ptoseis" + getDateTIme();
124     String state = Environment.getExternalStorageState();
125     if (Environment.MEDIA_MOUNTED.equals(state)) {
126         if (checkPermission()) {
127             File sdcard = Environment.getExternalStorageDirectory();
128             File dir = new File(sdcard.getAbsolutePath() + "/text/");
129             dir.mkdir();
130             File file = new File(dir, filename);
131             File file1 = new File(dir, fileptoseis);
132             try {
133                 os = new FileOutputStream(file, true);
134                 os.write("time,nsample,x,y,z,proj,projMean,deltaProj,var,std,
135                     slack,threshold,gNeg,gPos\n".getBytes());
136                 fos = new FileOutputStream(file1, true);
137             } catch (IOException e) {
138                 e.printStackTrace();
139             } else {
140                 requestPermission();
141             }
142         }
143         mFusedLocationClient = LocationServices.
144             getFusedLocationProviderClient(this);
145         getLastLocation();
146     }

```

```

146 public void sendSMS() {
147     try {
148         SmsManager smsmgr = SmsManager.getDefault();
149         String msg = "Xreiazomai_voithia!_syntetagmenes:_ " + location.
            getLatitude() + "_ " + location.getLongitude();
150         smsmgr.sendTextMessage(contactNumber, null, msg, null, null);
151         Toast.makeText(MainActivity.this, "SMS_Sent_Successfully", Toast.
            LENGTH_SHORT).show();
152     } catch (Exception e) {
153         Toast.makeText(MainActivity.this, "SMS_Failed_to_Send,_Please_try_
            again", Toast.LENGTH_SHORT).show();
154     }
155 }
156 public void savePrefs(View v) {
157     SharedPreferences.Editor editor = sharedPref.edit();
158     editor.putString("conNumber", contactNumber.getText().toString());
159     editor.putString("conEmail", contactEmail.getText().toString());
160     editor.putBoolean("checkSMS", checkBoxEmail.isChecked());
161     editor.putBoolean("checkEmail", checkBoxEmail.isChecked());
162     editor.apply();
163 }
164 private void getLastLocation() {
165     if (checkPermission()) {
166         if (isLocationEnabled()) {
167             mFusedLocationClient.getLastLocation().addOnCompleteListener(new
                OnCompleteListener<Location>() {
168                 @Override
169                 public void onComplete(@NonNull Task<Location> task) {
170                     location = task.getResult();
171                     if (location == null) {
172                         requestNewLocationData();
173                     } else {
174                         Log.i(TAG, "Latitude:" + location.getLatitude() + "
                            Longitude:_ " + location.getLongitude());
175                     }
176                 }
177             });
178         } else {
179             Toast.makeText(this, "Turn_on_location", Toast.LENGTH_LONG).show
                ();
180             Intent intent = new Intent(Settings.
                ACTION_LOCATION_SOURCE_SETTINGS);
181             startActivity(intent);
182         }
183     } else {
184         requestPermission();
185     }

```

```

186     }
187     private void requestNewLocationData() {
188         LocationRequest mLocationRequest = new LocationRequest();
189         mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
190             ;
191         mLocationRequest.setInterval(60000);
192         mLocationRequest.setFastestInterval(10000);
193         mFusedLocationClient = LocationServices.
194             getFusedLocationProviderClient(this);
195         mFusedLocationClient.requestLocationUpdates(mLocationRequest,
196             mLocationCallback, Looper.myLooper());
197     }
198     private LocationCallback mLocationCallback = new LocationCallback() {
199         @Override
200         public void onLocationResult(LocationResult locationResult) {
201             Location mLastLocation = locationResult.getLastLocation();
202             Log.i(TAG, "Latitude:␣" + mLastLocation.getLatitude() + "␣Longitude
203                 :␣" + mLastLocation.getLongitude());
204         }
205     };
206     public void startLog(View v) {
207         logging = true;
208         filename = "sensors" + getDateTIme();
209         fileptoseis = "ptoseis" + getDateTIme();
210         grapseFile(filename, fileptoseis);
211     }
212     @Override
213     public void onSensorChanged(SensorEvent event) {
214         xText.setText("x:" + event.values[0]);
215         yText.setText("y:" + event.values[1]);
216         zText.setText("z:" + event.values[2]);
217         nsamples= nsamples + 1.0;
218
219         if (logging) {
220             String stringSensors = getDateTIme() + "," + nsamples + "," + event
221                 .values[0] + "," + event.values[1] + "," + event.values[2]+",";
222             try {
223                 os.write(stringSensors.getBytes());
224             } catch (IOException e) {
225                 e.printStackTrace();
226             }
227         }
228
229         dian[0]=event.values[0];
230         dian[1]=event.values[1];

```

```
228     dian[2]=event.values[2];
229     calcMean(dian,nsamples);
230     pcaCheck(dian, mean, nsamples);
231     Log.i(TAG, "nsample:"+ nsamples );
232 }
233
234 @Override
235 public void onAccuracyChanged(Sensor sensor, int accuracy) {
236     //bleh
237 }
238 public void grapseFile(String filename, String fileptoseis) {
239     String state = Environment.getExternalStorageState();
240     if (Environment.MEDIA_MOUNTED.equals(state)) {
241         if (checkPermission()) {
242             File sdcard = Environment.getExternalStorageDirectory();
243             File dir = new File(sdcard.getAbsolutePath() + "/text/");
244             dir.mkdir();
245             File file = new File(dir, filename);
246             File file1 = new File(dir, fileptoseis);
247             try {
248                 os = new FileOutputStream(file, true);
249                 os.write("time,nsample,x,y,z,proj,projMean,deltaProj,var,std,
250                     slack,threshold,gNeg,gPos\n".getBytes());
251                 fos = new FileOutputStream(file1, true);
252             } catch (IOException e) {
253                 e.printStackTrace();
254             } else {
255                 requestPermission();
256             }
257         }
258     }
259
260 public void stopLog(View v) {
261     logging = false;
262     try {
263         os.flush();
264         os.close();
265     } catch (IOException e) {
266         e.printStackTrace();
267     }
268     // Min apenergopoiisete tha steleni SMS
269     sendSMS();
270 }
271 /* public void updateFile(View v) {
272     String result = new String();
273     switch (v.getId()) {
```

```

274     case R.id.buttonTP:
275         result = nsamples + "," + getDateTime() + ",tp";
276         break;
277     case R.id.buttonFP:
278         result = nsamples + "," + getDateTime() + ",fp";
279         break;
280     case R.id.buttonFN:
281         result = nsamples + "," + getDateTime() + ",fn";
282         break;
283     }
284 } */
285 public void initializeMean(View v) {
286     // To be implemented
287 }
288 public String getDateTime() {
289     Calendar c = Calendar.getInstance();
290     SimpleDateFormat df = new SimpleDateFormat("yyyyMMddHHmmss");
291     String formattedDate = df.format(c.getTime());
292     return formattedDate;
293 }
294 public void updateVars(View v) {
295     drift = Double.valueOf(driftText.getText().toString());
296     lamda = Double.valueOf(lamdaText.getText().toString());
297     threshold = Double.valueOf(thresholdText.getText().toString());
298 }
299 public void calcMean(double[] dian, double nsamples) {
300     mean[0] = (meanOld[0]*(nsamples-1) + dian[0])/nsamples;
301     mean[1] = (meanOld[1]*(nsamples-1) + dian[1])/nsamples;
302     mean[2] = (meanOld[2]*(nsamples-1) + dian[2])/nsamples;
303     meanOld[0] = mean[0];
304     meanOld[1] = mean[1];
305     meanOld[2] = mean[2];
306 }
307 public void pcaCheck(double[] dian, double[] mean, double nsamples) {
308     dataMeanSubtracted.set(0,0,dian[0]-mean[0]);
309     dataMeanSubtracted.set(0,1,dian[1]-mean[1]);
310     dataMeanSubtracted.set(0,2,dian[2]-mean[2]);
311     SimpleMatrix variance = dataMeanSubtracted.transpose().mult(
312         dataMeanSubtracted);
313     ipcav=ipcavOld.scale((nsamples-1-lamda)/nsamples).plus(ipcavOld.
314         divide(ipcavOld.normF()).mult(variance.scale((1+lamda)/nsamples))
315         );
316     ipcavOld = ipcav;
317     provoli = ipcav.mult(dataMeanSubtracted.transpose()).get(0);
318     provoliMean = (provoliMean*(nsamples-1) + provoli)/nsamples;
319     if (logging) {
320         try {

```



```

318         os.write((provoli + "," + provoliMean + "," + (provoli -
319             provoliMean)+",").getBytes());
320     } catch (IOException e) {
321         e.printStackTrace();
322     }
323     cusum(provoli, provoliMean);
324     //     }
325 }
326 public void cusum(double provoli, double provoliMean) {
327     currVariance = (currVariance*(nsamples-1) + Math.pow((provoli -
328         provoliMean),2))/(nsamples);
329     double currStanDev = Math.sqrt(currVariance);
330     double slack = currStanDev * drift;
331     double currThreshold = currStanDev * threshold;
332
333     gPos = Math.max(0, gPosOld + provoli - provoliMean - slack);
334     gPosOld = gPos;
335
336     gNeg = Math.min(0, gNegOld + provoli - provoliMean + slack);
337     gNegOld = gNeg;
338
339     if (logging){
340         try {
341             os.write((currVariance + "," + currStanDev + "," + slack + "," +
342                 currThreshold + "," + gNeg + "," + gPos).getBytes());
343             os.write('\n');
344         }catch (IOException e) {
345             e.printStackTrace();
346         }
347     }
348     if (nsamples > 80) {
349         if ((gPos > currThreshold) || ((Math.abs(gNeg)) > currThreshold)) {
350             fall = true;
351             try {
352                 fos.write(("ptos!_sto_!nsample"+ nsamples).getBytes());
353                 fos.write('\n');
354                 //             v.vibrate(500);
355             }catch (IOException e) {
356                 e.printStackTrace();
357             }
358             Log.i(TAG, "ptos!" + nsamples);
359             if (checkBoxSMS.isChecked()) {
360                 sendSMS();
361             }
362             resetCuSum();

```

```
362     }
363 }
364 }
365
366 public void resetCuSum() {
367     gPos = 0.0;
368     gPosOld = 0.0;
369     gNeg = 0.0;
370     gNegOld = 0.0;
371 }
372 // Elegxos gia perimission gia eggrafi sto external Storage
373 private boolean checkPermission() {
374     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.
375         WRITE_EXTERNAL_STORAGE) == PackageManager.PERMISSION_GRANTED &&
376         ActivityCompat.checkSelfPermission(this, Manifest.permission.
377             ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED &&
378         ActivityCompat.checkSelfPermission(this, Manifest.permission.
379             ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED &&
380         ActivityCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS)
381             == PackageManager.PERMISSION_GRANTED){
382         return true;
383     } else {
384         return false;
385     }
386 }
387 // Zitisi permission gia eggrafi sto External Storage
388 private void requestPermission() {
389     ActivityCompat.requestPermissions(MainActivity.this, new String[]{
390         Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.
391             ACCESS_COARSE_LOCATION, Manifest.permission.ACCESS_FINE_LOCATION,
392             Manifest.permission.SEND_SMS}, PERMISSION_REQUEST_CODE);
393 }
394 // Elegxos Apantisis stin zitisi permission
395 @Override
396 public void onRequestPermissionsResult(int requestCode, String
397     permissions[], int[] grantResults) {
398     switch (requestCode) {
399         case PERMISSION_REQUEST_CODE:
400             if (grantResults.length > 0 && grantResults[0] == PackageManager.
401                 PERMISSION_GRANTED) {
402                 Log.e("value", "Permission Granted, Now you can use local drive.");
403             } else {
404                 Log.e("value", "Permission Denied, You cannot use local drive.");
405             }
406             break;
407     }
408 }
```

```
399 }
400 private boolean isLocationEnabled() {
401     LocationManager locationManager = (LocationManager) getSystemService(
402         Context.LOCATION_SERVICE);
403     return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
404         || locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
405 }
```

Listing 8.1: Κώδικας Java