

Ιόνιο Πανεπιστήμιο Τμήμα Πληροφορικής



Πτυχιακή Εργασία

Κινητή Εφαρμογή Αποτυπώματος Άνθρακα με Εκτίμηση
του Μέσου Μετακίνησης

Γεώργιος Οδ. Μαντέλλος

Π2016149

Επιβλέπουσα Καθηγήτρια: Ελένη Χριστοπούλου

Κέρκυρα, Ιούνιος, 2020

Επιβλέπουσα καθηγήτρια:

Ελένη Χριστοπούλου, Εργαστηριακό Διδακτικό Προσωπικό
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή:

Κωνσταντίνος Οικονόμου, Αναπληρωτής Καθηγητής
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Μιχαήλ Στεφανιδάκης, Επίκουρος Καθηγητής
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Ελένη Χριστοπούλου, Εργαστηριακό Διδακτικό Προσωπικό
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω την επιβλέπουσα, κα. Ελένη Χριστοπούλου, για τη βοήθεια και την καθοδήγηση που παρείχε καθ' όλη τη διάρκεια εκπόνησης αυτής της πτυχιακής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Θεμιστοκλή Έξαρχο, ο οποίος συμμετείχε στην καθοδήγηση και έπαιξε σημαντικό ρόλο στη διαδικασία δημιουργίας του μοντέλου αυτόματης εκτίμησης.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, η οποία στάθηκε αρωγός σε όλη τη διάρκεια της φοίτησής μου.

Περίληψη

Πολλές φορές, κατά τη διάρκεια της ημέρας, πραγματοποιούμε μετακινήσεις, χωρίς να αντιλαμβανόμαστε τις συνέπειες της ρύπανσης που προκαλεί το μέσο της μετακίνησής μας. Αυτό, έχει ως αποτέλεσμα να παράγουμε πολύ περισσότερο διοξείδιο του άνθρακα, από όσο θα μας αναλογούσε. Για παράδειγμα, επιλέγουμε να μεταφερθούμε με το αυτοκίνητο αντί του αστικού λεωφορείου ή επιλέγουμε το αστικό λεωφορείο αντί της μετακίνησής μας με τα πόδια. Όλες αυτές οι επιλογές μας, μπορεί να φαίνονται ασήμαντες, ωστόσο, όταν γίνονται σε καθημερινή βάση και από πολλούς ανθρώπους, τότε τα αποτελέσματα μπορεί να είναι καταστροφικά για το περιβάλλον. Εάν, δεν μπορέσουμε να αντιληφθούμε έγκαιρα τις συνέπειες των συγκεκριμένων πράξεών μας, ίσως να είναι πολύ αργά. Ένας τρόπος για να συνειδητοποιήσουμε αν όντως οι μετακινήσεις μας είναι περιβαλλοντικά επιβλαβείς, είναι να τις καταγράψουμε και να υπολογίζουμε το ποσοστό ρύπανσης που προκαλούμε. Επιπλέον, όλο αυτό πρέπει να γίνεται σε καθημερινή βάση και για μεγάλα χρονικά διαστήματα, προκειμένου να υπάρχει μια σαφής εικόνα των εκπομπών του διοξειδίου του άνθρακα.

Αυτό λοιπόν το πρόβλημα, καλείται να λύσει η εφαρμογή RecognitionApp, που αναπτύξαμε. Συγκεκριμένα, η λειτουργία της είναι να καταγράφει καθημερινά τις μετακινήσεις μας, καθώς και το μέσο που χρησιμοποιήθηκε για την κάθε μια. Αυτό όμως, θα μπορούσε να επιτευχθεί και αν απλώς ο κάθε ένας από εμάς στο τέλος της ημέρας, κατέγραφε όλες τις μετακινήσεις που έκανε και υπολόγιζε χειροκίνητα, το αποτύπωμα του άνθρακα που θα προκαλούνταν από αυτές. Συνεπώς, τι νόημα έχει η ύπαρξη της συγκεκριμένης εφαρμογής;

Η χρησιμότητα ύπαρξης αυτής της εφαρμογής, έγκειται στο γεγονός πως διαχειρίζεται τα πάντα αυτόματα και αυτόνομα. Δηλαδή, δε χρειάζεται ο χρήστης να δηλώνει χειροκίνητα το μέσο της μετακίνησής του. Η εφαρμογή, μπορεί να αντιλαμβάνεται το μέσο των μετακινήσεων, να τις συγκεντρώνει και να υπολογίζει το αποτύπωμα του άνθρακα που αναλογεί κατά χρήστη, καθώς και να εμφανίζει λεπτομερή στατιστικά δεδομένα. Κατ' αυτόν τον τρόπο, μπορούμε να έχουμε επίγνωση του ποσοστού ρύπανσης που προκαλείται από τις μετακινήσεις μας και αν κριθεί απαραίτητο, να τις προσαρμόσουμε ως προς το μέσο και να τις βελτιώσουμε.

Συνοψίζοντας, μπορεί να είμαστε σε θέση να μετακινούμαστε με οποιονδήποτε τρόπο θέλουμε, αλλά η μετακίνηση μας αυτή, θα πρέπει να πραγματοποιείται με σεβασμό, τόσο ως προς το περιβάλλον, όσο και ως προς τους συνανθρώπους μας.

Περιεχόμενα

1. Εισαγωγή	7
1.1 Σκοπός	7
1.2 Στόχοι	7
1.3 Συνεισφορά - Προκλήσεις	7
1.4 Διάρθρωση	8
2. Βιβλιογραφική Επισκόπηση	10
2.1 Αναγνώριση της ανθρώπινης δραστηριότητας	10
2.2 Αναγνώριση του τύπου μετακίνησης χρησιμοποιώντας αισθητήρες χαμηλής κατανάλωσης ενέργειας έξυπνων κινητών τηλεφώνων	15
2.3 Εφαρμογές αποτυπώματος άνθρακα	15
3. Η Εφαρμογή για την Αυτόματη Εκτίμηση του Μέσου Μετακίνησης	17
3.1 Μεθοδολογία	17
3.2 Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν	19
3.3 Διαδικασία ανάπτυξης της εφαρμογής	20
3.3.1 Διαδικασία αυτόματης εκτίμησης του μέσου μετακίνησης μέσω της εφαρμογής	20
3.3.2 Προγραμματιστική ανάλυση	22
3.3.3 Προβλήματα που παρουσιάστηκαν και πώς επιλύθηκαν	23
3.4 Σύγκριση της εφαρμογής με αντίστοιχες	24
4. Η Εφαρμογή για τη Συλλογή των Δεδομένων	26
4.1 Ανάπτυξη της εφαρμογής για τη συλλογή δεδομένων	26
4.1.1 Προγραμματιστική ανάλυση	27
4.1.2 Προβλήματα που παρουσιάστηκαν και πώς επιλύθηκαν	29
4.1.3 Διαδικασία δοκιμής του λογισμικού	31
5. Τα Δεδομένα και το Μοντέλο για την Αυτόματη Εκτίμηση του Μέσου Μετακίνησης	32
5.1 Αξιολόγηση των δεδομένων	32
5.2. Επιδιόρθωση των δεδομένων	33
5.3 Υπολογισμός των στατιστικών μετρικών	33
5.4. Εξαγωγή του αρχείου δεδομένων (Dataset)	39
5.5. Εύρεση του μοντέλου μηχανικής μάθησης	39
5.5.1 Οι αλγόριθμοι που δοκιμάστηκαν και τα αποτελέσματά τους	39
5.5.2 Η επιλογή του αλγόριθμου PART	46
5.6. Διαδικασία παραγωγής κανόνων για την κινητή εφαρμογή αυτόματης εκτίμησης του μέσου μετακίνησης	47
5.7 Σύγκριση με άλλες ερευνητικές εργασίες	48
6. Υλοποίηση και Οθόνες	57

6.1 Σενάριο χρήσης και ενδεικτικές οθόνες για την εφαρμογή αυτόματης εκτίμησης του μέσου μετακίνησης	57
6.1.1 Σκούρο θέμα εμφάνισης	63
6.1.2 Φωτεινό θέμα εμφάνισης	64
6.2 Σενάριο χρήσης και ενδεικτικές οθόνες για την εφαρμογή συλλογής των δεδομένων	65
7. Συμπεράσματα	69
7.1 Ερευνητικά Συμπεράσματα	69
7.2 Προγραμματιστικά Συμπεράσματα	69
7.3 Πιθανές κατευθύνσεις για μελλοντική συνέχεια της έρευνας	70
Βιβλιογραφία	72

Πίνακας πινάκων

Πίνακας 1. Αποτελέσματα αλγόριθμου PART για Unbalanced και Balanced	40
Πίνακας 2. Αναλυτικά αποτελέσματα αλγόριθμου PART για Unbalanced	41
Πίνακας 3. Αναλυτικά αποτελέσματα αλγόριθμου PART για Balanced	42
Πίνακας 4. Αποτελέσματα αλγόριθμου J48 για Unbalanced και Balanced	42
Πίνακας 5. Αναλυτικά αποτελέσματα αλγόριθμου J48 για Unbalanced	43
Πίνακας 6. Αναλυτικά αποτελέσματα αλγόριθμου J48 για Balanced	43
Πίνακας 7. Αναλυτικά αποτελέσματα αλγόριθμου KNN για Unbalanced	44
Πίνακας 8. Αναλυτικά αποτελέσματα αλγόριθμου KNN για Balanced	45
Πίνακας 9. Αναλυτικά αποτελέσματα αλγόριθμου Naïve Bayes για Unbalanced	45
Πίνακας 10. Αναλυτικά αποτελέσματα αλγόριθμου Naïve Bayes για Balanced	46
Πίνακας 11. Οι κατηγορίες της πρώτης συγκρινόμενης έρευνας [28]	49
Πίνακας 12. Οι μετρικές που χρησιμοποιήθηκαν από την πρώτη συγκρινόμενη έρευνα [28]	50
Πίνακας 13. Οι δραστηριότητες που πραγματοποιήθηκαν από τη δεύτερη συγκρινόμενη έρευνα [29]	51
Πίνακας 14. Τα χαρακτηριστικά που αποκτήθηκαν από τη δεύτερη συγκρινόμενη έρευνα [29]	52
Πίνακας 15. Τα αποτελέσματα που εξήχθησαν από τη δεύτερη συγκρινόμενη έρευνα [29]	53
Πίνακας 16. Τα χαρακτηριστικά που εξήχθησαν από την τρίτη συγκρινόμενη έρευνα [33]	55
Πίνακας 17. Τα αποτελέσματα των αλγορίθμων από την τρίτη συγκρινόμενη έρευνα για τα δεδομένα “όλες οι πληροφορίες του ταξιδιού” [33]	56
Πίνακας 18. Τα αποτελέσματα των αλγορίθμων από την τρίτη συγκρινόμενη έρευνα για τα δεδομένα “όλες οι πληροφορίες του ταξιδιού εκτός του χρόνου που συμπεριλάμβανε τη χρήση κινητού και στάσεων” [33]	56

1. Εισαγωγή

Σε αυτή την ενότητα θα αναλυθεί ο κύριος σκοπός της παρούσας πτυχιακής εργασίας, οι προσδοκώμενοι στόχοι, η συνεισφορά της εργασίας στην έρευνα προς την αυτόματη αναγνώριση της ανθρώπινης δραστηριότητας καθώς και οι προκλήσεις που παρουσιάστηκαν. Τέλος, θα παρουσιαστεί και η διάρθρωση της εργασίας.

1.1 Σκοπός

Το θέμα αυτής της πτυχιακής εργασίας, ήταν η δημιουργία μιας κινητής εφαρμογής που ως σκοπό θα είχε, την αυτόματη εκτίμηση του μέσου μετακίνησης ενός χρήστη, προκειμένου να υπολογίζει το προσωπικό του αποτύπωμα άνθρακα. Η αυτόματη αυτή εκτίμηση, θα γινόταν με τη χρήση τριών αισθητήρων του κινητού. Το επιταχυνσιόμετρο, το γυροσκόπιο και το μαγνητόμετρο. Έπειτα, αξιοποιώντας τα δεδομένα που θα συλλέγονταν, θα δημιουργούταν ένα μοντέλο τεχνητής νοημοσύνης. Το μοντέλο αυτό, θα ήταν ικανό να καταλαβαίνει αυτόματα το μέσο της μετακίνησης ενός χρήστη.

1.2 Στόχοι

Τα αποτελέσματα που προσδοκούμε μέσω της συγκεκριμένης πτυχιακής εργασίας, είναι ουσιαστικά, να εξετάσουμε το ενδεχόμενο πραγματοποίησης ενός τέτοιου σεναρίου. Αν δηλαδή, είναι εφικτό να προβλεφθεί το μέσο της μετακίνησης ενός ατόμου χρησιμοποιώντας τρεις αισθητήρες. Ένας επιπλέον στόχος, είναι η εκτίμηση να πραγματοποιείται χωρίς τη χρήση του αισθητήρα GPS. Αυτό, διότι πρώτον, θέλουμε να εξετάσουμε αν μόνο η σχετική κίνηση ενός ατόμου, δηλαδή, η κίνηση που έχει το ανθρώπινο σώμα κατά τη διάρκεια μιας μετακίνησης, είναι ικανή ξεχωρίσει τους διαφορετικούς τρόπους μετακίνησης και δεύτερον, το γεγονός πως η χρήση του αισθητήρα GPS, έχει πολύ μεγάλη κατανάλωση μπαταρίας. Τι νόημα θα είχε μια εφαρμογή που εκτιμά αυτόματα το μέσο της μετακίνησης και ταυτόχρονα, σπαταλάει ένα τεράστιο μέρος της μπαταρίας του κινητού;

Οι τρόποι μετακίνησης που θα εξεταστούν, είναι το περπάτημα, το τρέξιμο, η μετακίνηση χρησιμοποιώντας το ποδήλατο, το αστικό λεωφορείο, το αυτοκίνητο και το τρένο.

1.3 Συνεισφορά - Προκλήσεις

Η συνεισφορά της πτυχιακής εργασίας στην έρευνα του HAR - Human Activity Recognition (Αναγνώριση Ανθρώπινης Δραστηριότητας) διαφαίνεται αρχικά, από τη

δημοσίευση [42] η οποία έγινε στα πλαίσια του συνεδρίου SEEDA-CECNSM 2020 με τίτλο “Human activity and transportation mode recognition using smartphone sensors”. Το συνέδριο SEEDA-CECNSM 2020 είναι το 5ο Συνέδριο Σχεδιασμού Αυτοματισμού, Μηχανικών Υπολογιστών, Δικτύων Υπολογιστών και Κοινωνικών Μέσων Νοτιοανατολικής Ευρώπης, τεχνικά συγχρηματοδοτείται από το IEEE Computational Intelligence Society (CIS) και θα διοργανωθεί από το Εργαστήριο Ανθρωπιστικής & Κοινωνικής Πληροφορικής.

Συνεχίζοντας, η συνεισφορά διαφαίνεται και από τη μεθοδολογία που ακολουθήθηκε για την αυτόματη εκτίμηση του μέσου σε πραγματικό χρόνο. Πιο συγκεκριμένα, οι περισσότερες από τις μέχρι τώρα έρευνες, έχουν μελετήσει την κατηγοριοποίηση της ανθρώπινης δραστηριότητας βασισμένοι σε έναν ή δυο αισθητήρες. Το επιταχυνσιόμετρο και το γυροσκόπιο. Οι μελέτες αυτές, έχουν δείξει πως κάτι τέτοιο είναι απόλυτα εφικτό και η ακρίβεια που παρέχεται είναι σχετικά μεγάλη. Ωστόσο, οι έρευνες αυτές, εστίαζαν στο γεγονός της κατηγοριοποίησης των δεδομένων και όχι στην αναγνώρισή τους σε πραγματικό χρόνο. Ωστόσο, δικός μας σκοπός εκτός από τη σωστή κατηγοριοποίηση, ήταν και η αναγνώριση σε πραγματικό χρόνο, των μετακινήσεων ενός χρήστη. Για αυτό λοιπόν τον σκοπό, στην έρευνά μας χρησιμοποιήσαμε και έναν τρίτο αισθητήρα. Τον αισθητήρα του μαγνητόμετρου. Θα μπορούσε να αναρωτηθεί βέβαια κάποιος, ποιος ο λόγος της προσθήκης ενός ακόμη αισθητήρα από τη στιγμή που οι ήδη υπάρχοντες είναι απόλυτα ικανοί για την κατηγοριοποίηση των δραστηριοτήτων.

Στη δική μας περίπτωση, εκτός από τις τυπικές φυσικές δραστηριότητες, δηλαδή το περπάτημα, θέλαμε και να αναγνωρίζουμε τις μετακινήσεις με όχημα και μάλιστα, να είμαστε σε θέση να ξεχωρίσουμε και τους διαφορετικούς τύπους των οχημάτων. Πιο συγκεκριμένα, τα οχήματα που θέλαμε να συμπεριλάβουμε ήταν το ποδήλατο, το αυτοκίνητο, το αστικό λεωφορείο και το τρένο. Σε αυτού του είδους τις μετακινήσεις, είναι σημαντικό να γνωρίζουμε και τη χρονική διάρκεια των ενδιάμεσων καταστάσεων, δηλαδή, των στάσεων και των στροφών κατά τη διάρκεια μιας μετακίνησης. Το μαγνητόμετρο, ήταν απόλυτα ικανό να καταγράψει αυτή τη “διάρκειά” στις καταστάσεις αυτές. Έτσι λοιπόν, συμπεριλαμβάνοντας αυτούς τους τρεις αισθητήρες, καταφέραμε να πραγματοποιήσουμε αναγνώριση δραστηριότητας σε πραγματικό χρόνο με εξαιρετικά μεγάλη ακρίβεια.

1.4 Διάρθρωση

Σε αυτό το σημείο θα αναλυθεί η δομή της πτυχιακής εργασίας. Συνεχίζοντας την ανάγνωση λοιπόν, στην ενότητα 2, υπάρχουν οι βιβλιογραφικές αναφορές, όπου αναλύεται ο ερευνητικός χώρος του αντικειμένου που πραγματεύεται η εργασία. Στην ενότητα 3 αναλύεται η μεθοδολογία, οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν καθώς και η διαδικασία ανάπτυξης της εφαρμογής για την αυτόματη αναγνώριση του μέσου μετακίνησης. Στην ενότητα 4 αναλύονται οι ίδιες

Θεματικές ενότητες από την ενότητα 3, με τη διαφορά ότι πρόκειται για την εφαρμογή συλλογής των δεδομένων που χρειάστηκαν, για να μπορέσουμε να αναπτύξουμε την εφαρμογή αυτόματης αναγνώρισης. Στην ενότητα 5, αναλύονται τα δεδομένα που συλλέχθηκαν από την εφαρμογή συλλογής δεδομένων καθώς και το μοντέλο που δημιουργήθηκε για να επιτευχθεί η αυτόματη αναγνώριση. Τέλος, στην ενότητα 6, παραθέτονται εικόνες και από τις δυο εφαρμογές μαζί με κάποια πιθανά σενάρια χρήσης τους. Τέλος, στην ενότητα 7, βρίσκονται τα συμπεράσματα της εργασίας.

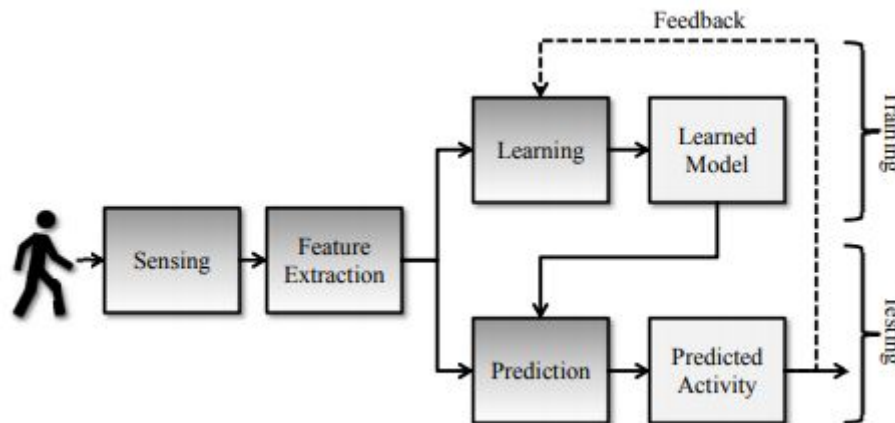
2. Βιβλιογραφική Επισκόπηση

Σε αυτή την ενότητα παρουσιάζεται η βιβλιογραφική επισκόπηση που μελετήθηκε στο θέμα της αναγνώρισης της ανθρώπινης δραστηριότητάς, προκειμένου να είναι πλήρως κατανοητό προς τον αναγνώστη το θέμα που πραγματεύεται η εργασία.

2.1 Αναγνώριση της ανθρώπινης δραστηριότητας

Τα έξυπνα κινητά τηλέφωνα δημιουργούν νέες ερευνητικές ευκαιρίες για εφαρμογές με επίκεντρο τον άνθρωπο, όπου ο χρήστης είναι μια πλούσια πηγή πληροφοριών και το τηλέφωνο, είναι το εργαλείο ανίχνευσης. Οι πιο πρόσφατες συσκευές διαθέτουν ενσωματωμένους αισθητήρες όπως μικρόφωνα, διπλές κάμερες, επιταχυνσιόμετρα, γυροσκόπια κ.λπ. Η χρήση των έξυπνων κινητών τηλεφώνων με αδρανειακούς αισθητήρες είναι μια εναλλακτική λύση για την ΑΑΔ (Αναγνώριση της ανθρώπινης δραστηριότητας). Αυτές οι συσκευές παρέχουν μια ευέλικτη, προσιτή και αυτόνομη λύση για την αυτόματη και διακριτική παρακολούθηση των δραστηριοτήτων της καθημερινής ζωής, ενώ παράλληλα, παρέχουν υπηρεσίες τηλεφωνίας. Κατά συνέπεια, τα τελευταία χρόνια, έχουν προταθεί ορισμένα έργα που στοχεύουν στην κατανόηση της ανθρώπινης συμπεριφοράς με τη χρήση των έξυπνων κινητών τηλεφώνων: για παράδειγμα στο [1], μία από τις πρώτες προσεγγίσεις για την εκμετάλλευση ενός έξυπνου κινητού τηλεφώνου με λειτουργικό σύστημα Android για τη ΑΑΔ που χρησιμοποιεί τα ενσωματωμένα τριαξονικά επιταχυνσιόμετρα. Επιπλέον, παρόμοιες μελέτες έχουν παρουσιαστεί και στο [2, 3]. Αναμένονται ακόμη βελτιώσεις σε θέματα όπως είναι η σύμπτυξη πολλών αισθητήρων μαζί για καλύτερη ταξινόμηση της ΑΑΔ και για την τυποποίηση των μετρικών αξιολόγησης της απόδοσης [4]. Επιπλέον, για την επαρκή σύλληψη της κίνησης του ανθρώπινου σώματος, απαιτούνται έως και 15Hz καθώς το 99% της ενέργειας του, περιέχεται κάτω από αυτό το όριο [5]. Μια γενική αναπαράσταση της διαδικασίας αναγνώρισης της ανθρώπινης δραστηριότητας, συμπεριλαμβανομένων των κύριων συστατικών της, απεικονίζεται στο Σχήμα 1. Επιπλέον, τα παραδοσιακά συστήματα ΑΑΔ λειτουργούν συνήθως σε βάση προώθησης, επομένως η εκμάθηση εκτελείται εκτός σύνδεσης, μόνο μία φορά και δεν υπάρχει περαιτέρω ανατροφοδότηση στο σύστημα. Αυτό είναι χρήσιμο σε περιπτώσεις όπου η τα δεδομένα δεν αλλάζουν με την πάροδο του χρόνου ή το σύστημα είναι ανεξάρτητο. Διαφορετικά, συνιστώνται προσαρμοσμένες μέθοδοι όπως η σταδιακή διαδικτυακή μάθηση ή, η μάθηση μέσω μεταφοράς [6], αλλά εξαρτώνται από την αύξηση του υπολογιστικού φορτίου. Έχουν προταθεί προσεγγίσεις στη βιβλιογραφία για την εκτέλεση του ΑΑΔ: Το έργο που παρουσιάστηκε στο [7] ήταν πρωτοπόρος στην ανάπτυξη μιας μεθόδου για την ανίχνευση ενός συνόλου δραστηριοτήτων καθημερινής ζωής που χρησιμοποιούν

πέντε επιταχυνσιόμετρα που φοριούνται στο σώμα και χρησιμοποιούν γνωστούς ταξινομητές μηχανικής μάθησης.



Σχήμα 1. Διαδικασία αναγνώρισης της ανθρώπινης δραστηριότητας [31]

Ο τρόπος με τον οποίο γίνεται η απόκτηση των δεδομένων είναι επίσης μια σημαντική πτυχή της ΑΑΔ. Θα πρέπει να αναπαράγεται όσο το δυνατόν πλησιέστερα στις πραγματικές συνθήκες της κίνησης. Επομένως, ελεγχόμενα πειράματα μπορούν να διεξαχθούν σε εργαστηριακές συνθήκες με στόχο την προσομοίωση των φυσικών ρυθμίσεων (ημι-φυσιοκρατικά περιβάλλοντα), ωστόσο, μη ελεγχόμενα πειράματα μπορούν επίσης να διεξαχθούν, αλλά, απαιτούν τη μεταγενέστερή επεξεργασία των δεδομένων που συλλέχθηκαν. Οι ελλείψεις στον σχεδιασμό των συστημάτων ΑΑΔ μπορεί να οφείλονται σε ζητήματα όπως είναι ο θόρυβος του αισθητήρα, ο ίδιος ο αισθητήρας ανίχνευσης ή ακόμη και η θέση του σε σχέση με το ανθρώπινο σώμα κ.λ.π. Το τελευταίο συνδέεται με την απόδοση του συστήματος όπως παρουσιάζεται στο [8, 9] όπου αξιολογήθηκαν διαφορετικές θέσεις αισθητήρων για τον καθορισμό των ιδανικών θέσεων μέσω της χρήσης φορετών επιταχυνσιόμετρων.

Η μείωση των χαρακτηριστικών για την αναγνώριση της δραστηριότητας, μπορεί να μειώσει σε σημαντικό και το αποτέλεσμα του αλγορίθμου που θα χρησιμοποιηθεί. Επομένως, πολλές πτυχές πρέπει να ληφθούν υπόψη για την επιλογή τους. Μια παραδοσιακή προσέγγιση στην ΑΑΔ είναι ο κατακερματισμός των σημάτων του αισθητήρα σε χρονικά παράθυρα με σταθερό μήκος, το οποίο εξαρτάται από την εφαρμογή. Για παράδειγμα, για την αναγνώριση των μεταβάσεων του ανθρώπινου σώματος από όρθιος σε καθιστός ή από περπάτημα σε τρέξιμο, απαιτούνται σύντομα χρονικά παράθυρα. Ωστόσο, άλλες πολύπλοκες δραστηριότητες, όπως διαταραχές κίνησης (π.χ. ανίχνευση καταστάσεων ON-OFF στους ασθενείς με νόσο του Parkinson (PD) [10]) ενδέχεται να απαιτούν μεγαλύτερες περιόδους διάρκειας παραθύρου για την εξασφάλιση βεβαιότητας σχετικά με την ανίχνευση μιας συγκεκριμένης κατάστασης. Είναι επίσης συνηθισμένο να επιλέγονται επικαλυπτόμενα παράθυρα (συνήθως 50% επικάλυψη [11]), καθώς με

αυτόν τον τρόπο είναι δυνατόν να γίνουν αντιληπτά τα χρονικά αυτά συμβάντα. Υπάρχουν πολλά χαρακτηριστικά που χρησιμοποιούνται συνήθως στη βιβλιογραφία [12] τα οποία λαμβάνονται κυρίως από τους τομείς χρόνου και της συχνότητας, παρόλο που ορισμένες άλλες εναλλακτικές περιλαμβάνουν τους συντελεστές μετασχηματισμού κυμάτων που επιτρέπουν συνδυασμένες αναπαραστάσεις σήματος της χρονικής συχνότητας. Μηχανισμοί για τη μείωσή του αριθμού των χαρακτηριστικών αρχίζουν να διαφαίνονται, όπως για η διαδικασία επιλογής χαρακτηριστικών όπου τα χαρακτηριστικά αξιολογούνται με διαφορετικές μετρικές για τον καθορισμό και την επιλογή των πιο ενημερωτικών υπό ομάδων τους, ή, για τη διαδικασία επιλογής των χαρακτηριστικών όταν η διάσταση του διανύσματος χαρακτηριστικών μπορεί να μειωθεί εκτελώντας μετασχηματισμούς μεταξύ των λειτουργιών είτε μέσω γραμμικών ή μη γραμμικών μεθόδων. Μερικά συστήματα HAR που έχουν εξετάσει αυτούς τους μηχανισμούς είναι το [13] και το [14].

Η αυτόματη ταξινόμηση της ανθρώπινης δραστηριότητας μπορεί να μελετηθεί με τη χρήση της Μηχανικής Μάθησης, εφαρμόζοντας αλγορίθμους εποπτευόμενης μάθησης, ωστόσο, έχουν προταθεί ημι-εποπτευόμενες και μη εποπτευόμενες μέθοδοι [15, 16]. Τα μοντέλα συχνότητων και τα μοντέλα Bayes, έχουν καλυφθεί καλά σε όλη τη βιβλιογραφία της ΑΑΔ, περιλαμβάνουν μοντέλα πρόβλεψης όπως δέντρα δυαδικών αποφάσεων και ταξινομητές βάσει κατωφλίου [17, 18], γεωμετρικές προσεγγίσεις συμπεριλαμβανομένων των K-Nearest-Neighbours (KNN), των Artificial Neural Networks (ANN), των Support Vector Machines (SVM) [8, 22, 10] και πιθανοτικές μέθοδοι ταξινόμησης όπως για παράδειγμα οι ταξινομητές Naïve Bayes και Hidden Markov Models (HMM) [20, 21]. Δεν είναι απολύτως σαφές ποια προσέγγιση Μηχανικής Μάθησης έχει καλύτερη απόδοση για την ΑΑΔ καθώς πολλά από αυτά έχουν δείξει συγκρίσιμη απόδοση σε διαφορετικές έρευνες (π.χ. [22]). Η βέλτιστη μέθοδος ταξινόμησης είναι γενικά διαφορετική ανά εφαρμογή [23] και πτυχές όπως η απόδοση αναγνώρισης, η κατανάλωση ενέργειας, οι απαιτήσεις μνήμης και η υπολογιστική πολυπλοκότητα καθίστανται καθοριστικοί παράγοντες. Για παράδειγμα, τα δέντρα αποφάσεων θα μπορούσαν να προτιμηθούν όταν απαιτείται η ερμηνεία του μοντέλου και τα SVMs για εφαρμογές υψηλής απόδοσης. Η συνεχής έρευνα σε σχέση με τις υπολογιστικές μειώσεις του κόστους μελετήθηκε εκτενώς, όπως τροποποιημένες αποδοτικές υλοποιήσεις των αλγορίθμων μηχανικής μάθησης όπως προτείνεται και στο [24], όπου παρουσιάστηκε μια καθαρά αριθμητική προσέγγιση σταθερού σημείου για την ΑΑΔ χρησιμοποιώντας SVMs.

Η αξιολόγηση των αλγορίθμων ταξινόμησης ΑΑΔ πραγματοποιείται κυρίως μέσω της στατιστικής ανάλυσης των μοντέλων χρησιμοποιώντας τα διαθέσιμα πειραματικά δεδομένα. Η πιο συνηθισμένη μέθοδος είναι ο πίνακας σύγχυσης (confusion matrix) που επιτρέπει την αναπαράσταση της απόδοσης του αλγορίθμου προσδιορίζοντας σαφώς τους τύπους σφαλμάτων (ψευδώς θετικά και αρνητικά) και σωστά προβλεπόμενα δείγματα στα δεδομένα δοκιμής. Διάφορες μετρήσεις μπορούν επίσης να εξαχθούν από τα δεδομένα, όπως ακρίβεια μοντέλου, ακρίβεια,

ανάκληση και βαθμολογία F1 [12]. Τέλος, άλλα συγκριτικά μέτρα μεταξύ των αλγορίθμων είναι η ταχύτητα πρόβλεψης και η κατανάλωση μνήμης.

Στο παρελθόν έχουν προταθεί αρκετές προσεγγίσεις στη βιβλιογραφία για την αναγνώριση ανθρώπινων δραστηριοτήτων οι οποίες καλύπτουν διαφορετικούς τομείς εφαρμογών, όπως υγειονομική περίθαλψη, έξυπνα σπίτια, πανταχού παρόντες υπολογιστές, διαβίωση με υποβοήθηση περιβάλλοντος και παρακολούθηση και ασφάλεια. Αυτές οι προσεγγίσεις μπορούν να κατηγοριοποιηθούν σύμφωνα με πολλά διαφορετικά κριτήρια, όπως, ανά τύπο αισθητήρα, ο οποίος εξαρτάται από τα σήματα που μετρώνται (π.χ. αδρανειακά, με βάση την όραση και φυσιολογικά), από τη θέση του αισθητήρα, δηλαδή από την εξωτερική ανίχνευση όταν οι αισθητήρες βρίσκονται σε σταθερές θέσεις στο περιβάλλον και από τη φορητή ανίχνευση όταν είναι συνδεδεμένοι στο σώμα, με την αρχή της μοντελοποίησης, η οποία μπορεί να βασίζεται σε δεδομένα ή γνώση ανάλογα με το αν τα μοντέλα ΑΑΔ κατασκευάζονται δεδομένου προϋπάρχοντος συνόλου δεδομένων ή από την εκμετάλλευση προηγούμενων γνώσεων σχετικά με έναν συγκεκριμένο τομέα και με την προσέγγιση της μάθησης, η οποία μπορεί να είναι είτε εποπτευόμενη, ημι-εποπτευόμενη είτε χωρίς επίβλεψη. Τα φορητά συστήματα έχουν τραβήξει την προσοχή της ερευνητικής κοινότητας της ΑΑΔ λόγω της ευκολίας λήψης πληροφοριών δραστηριότητας (π.χ. κίνηση σώματος, θερμοκρασία και καρδιακός ρυθμός) απευθείας από τον χρήστη, διακριτικά και ουσιαστικά σε οποιαδήποτε τοποθεσία χωρίς την ανάγκη σταθερής υποδομής σε αντίθεση με συστήματα που βασίζονται στην όραση. Η εργασία που παρουσιάστηκε στο [27] ήταν πρωτοπόρος στην ανάπτυξη μιας προσέγγισης για την ταξινόμηση της καθημερινής δραστηριότητας χρησιμοποιώντας πέντε επιταχυνσιόμετρα που φοριούνται στο σώμα και χρησιμοποιώντας γνωστούς ταξινομητές τεχνητής νοημοσύνης. Έκτοτε, έχουν προταθεί και άλλες προσεγγίσεις [12] που στοχεύουν σε διαφορετικές εφαρμογές, για παράδειγμα, από ιατρική άποψη, έχουν παρουσιαστεί συστήματα παρακολούθησης για την ανίχνευση διαφορετικών χαρακτηριστικών σε ηλικιωμένους ασθενείς με Πάρκινσον όπως παράμετροι βάδισης, διαταραχές κίνησης και πτώσεις χρησιμοποιώντας τον αισθητήρα του επιταχυνσιόμετρου στο σώμα.

Πιο πρόσφατα, οι ερευνητικές προσπάθειες επικεντρώθηκαν στην εκμετάλλευση των smartphones για την ΑΑΔ. Οι εφαρμογές που βασίζονται στα έξυπνα κινητά τηλέφωνα προσφέρουν διάφορα οφέλη σε σύγκριση με άλλες γνωστές εναλλασσόμενες εναλλακτικές λύσεις της ΑΑΔ που χρησιμοποιούν συσκευές ειδικού σκοπού ή δίκτυα αισθητήρων που είναι συνδεδεμένα στο σώμα. Τα κύρια πλεονεκτήματά τους βασίζονται στην εύκολη φορητότητα της συσκευής, στην διακριτική αίσθηση που παρέχεται από τους ενσωματωμένους αισθητήρες και στην ισχύ επεξεργασίας των συσκευών που επιτρέπουν την εκτέλεση online ακριβών υπολογισμών. Έχουν ερευνηθεί αρκετές προσεγγίσεις που βασίζονται σε έξυπνα κινητά τηλέφωνα που χρησιμοποιούν διάφορους ενσωματωμένους

αισθητήρες (π.χ. επιταχυνσιόμετρο, γυροσκόπιο, GPS, μαγνητόμετρο) [12]. Οι προσεγγίσεις μηχανικής μάθησης που έχουν ήδη εφαρμοστεί για την αναγνώριση δραστηριοτήτων περιλαμβάνουν τους αλγορίθμους Naive Bayes, αλυσίδες Markov, Decision Trees και Support Vector Machines (SVMs). Τα SVMs παρέχουν έναν καλό συμβιβασμό μεταξύ της ακρίβειας και του χρόνου εκπαίδευσης, ενώ μετρούν επίσης με μια ποικιλία διαθέσιμων στο κοινό εργαλείων μάθησης για πειραματισμό, όπως το LIBSVM. Ωστόσο, ενώ χαρακτηρίζεται από πολλά ελκυστικά χαρακτηριστικά, ένα από τα κύρια μειονεκτήματα του SVM είναι στη φύση του naive two-class, που καθιστά τη γενίκευση σε προβλήματα πολλαπλών κατηγοριών (όπως στην τυπική περίπτωση της ΑΑΔ) όχι απλή. Έχουν διερευνηθεί διαφορετικές προσεγγίσεις για τη στόχευση αυτού του προβλήματος. Οι δύο πιο συχνά χρησιμοποιούμενες μέθοδοι είναι η One-Vs-All (OVA) και η One-Vs-One (OVO), όπου κάθε συγκεκριμένη κλάση συγκρίνεται με τη χρήση ενός δυαδικού ταξινομητή με τις υπόλοιπες κλάσεις όλες μαζί (OVA) είτε ανά μία (OVO) για τον προσδιορισμό της πιο πιθανής κλάσης για κάθε νέο δείγμα. Η απόδοση της προσέγγισης OVA είναι συγκρίσιμη με την ταξινόμηση OVO, όπως έχει επιβεβαιωθεί προηγουμένως στο. Επιπλέον, το παραγόμενο μοντέλο της μεθόδου OVA χρειάζεται λιγότερη μνήμη σε σύγκριση με τη μέθοδο OVO, δημιουργώντας ένα πλεονέκτημα λαμβάνοντας υπόψη τους περιορισμένους διαθέσιμους πόρους. Η εκμετάλλευση μοντέλων SVM για την ΑΑΔ σε έξυπνα κινητά τηλέφωνα, ωστόσο, απαιτεί αρκετές λειτουργίες κινητής υποδιαστολής ανά δευτερόλεπτο: παρά το γεγονός ότι δεν αποτελεί πρόβλημα από θεωρητική άποψη, αυτό θα μπορούσε να οδηγήσει σε αποφόρτιση της μπαταρίας μετά από μερικές ώρες συνεχούς λειτουργίας, κάνοντας αυτή την προσέγγιση ανέφικτη. Επιπλέον, οι περιορισμένες συσκευές προτιμώνται συνήθως για εφαρμογές ειδικού σκοπού εάν επιδεικνύουν παρόμοια απόδοση με τις παραδοσιακές μονάδες επεξεργασίας, καθώς το κόστος παραγωγής (και / ή απόκτησης) είναι γενικά χαμηλότερο. Σήμερα, έχει γίνει ιδιαίτερα ενδιαφέρον να επαναλάβουμε αυτές τις προσεγγίσεις και να τις εφαρμόσουμε στην ανάπτυξη εφαρμογών λογισμικού για φορητές συσκευές, όπως τα έξυπνα κινητά τηλέφωνα, που είναι ιδιαίτερα απαιτητικές όσον αφορά την κατανάλωση ενέργειας και τη διαχείριση πόρων συστήματος. Αυτή η μέθοδος σχεδιάστηκε για προβλήματα δυαδικής ταξινόμησης χρησιμοποιώντας αριθμητικό σταθερό σημείο στη φάση προώθησης του ταξινομητή SVM, με σκοπό, να επιτρέπεται η χρήση του σε συσκευές περιορισμένου υλικού. Τις τελευταίες δεκαετίες έχουν αφιερωθεί αρκετά έργα για την προσαρμογή των προσεγγίσεων τεχνητής νοημοσύνης σε συγκεκριμένες πλατφόρμες υλικού και, συγκεκριμένα, να αναλύσουν τα αποτελέσματα της κβαντοποίησης παραμέτρων στις φάσεις εκπαίδευσης και προώθησης. Τα κίνητρα για αυτές τις δραστηριότητες συνδέονται συνήθως με απαιτήσεις για συγκεκριμένες εφαρμογές, αλλά και με τη βασική αρχή της Θεωρίας της Στατιστικής Μάθησης (SLT) όπου πρέπει να αναζητήσουμε το ευκολότερο μοντέλο που σωστά ταξινομεί τα διαθέσιμα δεδομένα. Η εισαγωγή χώρων βασισμένων σε bit, αποφέρει ευρέως οφέλη στη μαθησιακή διαδικασία των

ταξινομητών (δηλαδή τάξεις συναρτήσεων όπου τα μοντέλα περιγράφονται μέσω περιορισμένων αριθμών bits). Αυτό οφείλεται στο γεγονός ότι η μείωση του αριθμού των bits επηρεάζει σε μεγάλο βαθμό την πολυπλοκότητα του χώρου υπόθεσης, το οποίο είναι ένα βασικό ζήτημα στη Μηχανική Μάθηση όπως υπογραμμίζεται στο. Εάν είμαστε σε θέση να μειώσουμε την πολυπλοκότητα του χώρου υπόθεσης χωρίς να επηρεάσουμε την ικανότητα του αλγορίθμου να μάθει τη λειτουργία με χαμηλό εμπειρικό σφάλμα, στην πράξη, είμαστε σε θέση να μάθουμε πιο αποτελεσματικά.

2.2 Αναγνώριση του τύπου μετακίνησης χρησιμοποιώντας αισθητήρες χαμηλής κατανάλωσης ενέργειας έξυπνων κινητών τηλεφώνων

Όσον αφορά την υλοποίηση του αλγορίθμου ταξινόμησης, ακολουθήσαμε μια διαδικτυακή προσέγγιση που βασίζεται στα έξυπνα κινητά τηλέφωνα, οπότε ο αλγόριθμος ταξινόμησης βασίζεται μόνο στους εσωτερικούς πόρους της συσκευής, γεγονός που προϋποθέτει ότι ο ίδιος ο αλγόριθμος πρέπει να είναι αρκετά απλός ώστε να μπορεί να εκτελείται σε αυτές τις πλατφόρμες [32]. Επιλέξαμε μια πολυτροπική αναγνώριση και οι πληροφορίες που χρησιμοποιούνται για την ταξινόμηση είναι ένας συνδυασμός των τιμών του επιταχυνσιόμετρου, του γυροσκοπίου και μαγνητόμετρου. Το μοντέλο μας επιτυγχάνει αποτελέσματα συγκρίσιμα με παρόμοιες λύσεις εκτός σύνδεσης ή λύσεις που χρησιμοποιούν αισθητήρες κατανάλωσης μπαταρίας [32], [33].

2.3 Εφαρμογές αποτύπωματος άνθρακα

Στόχος μας ήταν να δημιουργήσουμε μια εφαρμογή για έξυπνες κινητές συσκευές ώστε να ευαισθητοποιήσουμε τους χρήστες για το περιβάλλον και να καθοδηγήσουμε τους ανθρώπους να ακολουθήσουν έναν πιο βιώσιμο τρόπο ζωής παρακολουθώντας και μειώνοντας το αποτύπωμα άνθρακά τους. Παρόλο που υπάρχουν πολλές εφαρμογές για κινητές συσκευές που βοηθούν τους χρήστες να παρακολουθούν το αποτύπωμα άνθρακα τους, όπως το Oroeco [34], το GoGreen: Carbon Tracker [35] και το Carbon Footprint Calculator [36], όλες αυτές οι εφαρμογές ζητούν από τους χρήστες να εισάγουν πολλά δεδομένα σχετικά με την καθημερινή τους ρουτίνα ώστε να μπορέσουν υπολογίσουν μια εκτίμηση για αυτά τα δεδομένα.

Έτσι λοιπόν, συνδυάσαμε όλα τα παραπάνω με σκοπό να δημιουργήσουμε μια εφαρμογή για έξυπνα κινητά τηλέφωνα που να καταγράφει τις καθημερινές μεταφορές των χρηστών, καθώς και τα μέσα που χρησιμοποιούν για κάθε μία. Για παράδειγμα, το περπάτημα, το ποδήλατο, την οδήγηση αυτοκινήτου ή τη διαδρομή

με το δημόσιο λεωφορείο, ώστε να υπολογίσουμε το αποτυπώματος άνθρακά τους, βάσει αυτού. Η κύρια καινοτομία της εφαρμογής μας έγκειται στο γεγονός ότι διαχειρίζεται τα πάντα αυτόματα και απρόσκοπτα, επομένως ο χρήστης δε χρειάζεται να τροφοδοτεί με μη αυτόματο τρόπο τη μεταφορά του. Η εφαρμογή μπορεί να αναγνωρίσει τον τρόπο μεταφοράς αυτόματα και να υπολογίσει το αποτύπωμα άνθρακα του χρήστη, καθώς και να εμφανίσει λεπτομερή στατιστικά. Με αυτόν τον τρόπο, ο χρήστης μπορεί να γνωρίσει το ποσοστό ρύπανσης που προκαλείται από τα ταξίδια του και εάν είναι απαραίτητο, να τα προσαρμόσει ανάλογα ή και να τα βελτιώσει.

3. Η Εφαρμογή για την Αυτόματη Εκτίμηση του Μέσου Μετακίνησης

Σε αυτή την ενότητα θα εξεταστεί η μεθοδολογία ανάπτυξης της κινητής εφαρμογής για την αυτόματη εκτίμηση του μέσου μετακίνησης. Πιο συγκεκριμένα θα αναλυθεί το σκεπτικό ανάπτυξης, οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν, η διαδικασία συγγραφής του πηγαίου κώδικα, η προγραμματιστική ανάλυσή του, καθώς και τα προβλήματα που παρουσιάστηκαν και πως επιλύθηκαν.

3.1 Μεθοδολογία

Το σκεπτικό της ανάπτυξης και της υλοποίησης της πτυχιακής εργασίας αναλύεται παρακάτω. Αρχικά, ο σκοπός ήταν να καταγράψουμε το αποτύπωμα άνθρακα ενός χρήστη όσον αφορά τις μετακινήσεις του. Συνεπώς, θα έπρεπε να είμαστε σε θέση να αντιλαμβανόμαστε ποια είδος μετακίνησης πραγματοποιεί. Για παράδειγμα, μετακίνηση με τα πόδια, μετακίνηση με το λεωφορείο, μετακίνηση με το ποδήλατο ή με το αυτοκίνητο.

Ωστόσο, δε γίνεται να έχουμε την απαίτηση από τον χρήστη να καταγράφει χειροκίνητα κάθε μετακίνησή του, μιας και αυτό μπορεί να εγκυμονεί κινδύνους και λάθη που στην αρχή, ίσως να μην είναι εύκολο να γίνουν αντιληπτά. Ένα τέτοιο λάθος θα μπορούσε να είναι ότι ο χρήστης θα επέλεγε να κάνει ένα είδος δραστηριότητας, το οποίο είναι διαφορετικό από αυτό που πραγματικά ήθελε να κάνει και να μην το αντιληφθεί στην αρχή. Συνεπώς, να χρειαστεί να διαγράψει όλη την καταγραφή. Μπορεί αυτό να φαίνεται ασήμαντο για μια καταγραφή, αλλά σε βάθος χρόνου, ο αριθμός των καταγραφών αυξάνεται εκθετικά, συνεπώς, να μην έχει σαφή εικόνα των μετακινήσεων και του αποτυπώματος άνθρακα που εκπέμπει. Προφανώς, δεν μπορούμε να του αφήσουμε το δικαίωμα να αλλάζει το είδος της μετακίνησης μεταγενέστερα της καταγραφής, μιας και με αυτό τον τρόπο, θα μπορούσε να μεταποιήσει εντελώς τα δεδομένα και συνεπώς, το αποτύπωμα άνθρακα που εκπέμπει να μην αντιστοιχεί στο πραγματικό. Με αυτό τον τρόπο, η εφαρμογή θα έχανε εντελώς την αξιοπιστία της. Άρα, ο χρήστης θα πρέπει να μπορεί, μόνο να διαγράψει τα δεδομένα και όχι να τα μεταποιήσει. Αναλύοντας περαιτέρω τις περιπτώσεις χρήσης, ένας κίνδυνος που προκύπτει είναι ουσιαστικά, το προηγούμενο λάθος, με τη διαφορά ότι μπορεί να γίνει με εσκεμμένο σκοπό. Συνεπώς, κάποιος να πραγματοποιήσει μια δραστηριότητα αλλά να δηλώσει συνειδητά μια άλλη, προκειμένου να μη φαίνεται το πραγματικό του αποτύπωμα άνθρακα. Συνεχίζοντας την ανάλυση, ένας άλλος κίνδυνος που διαφαίνεται, είναι ότι ο χρήστης θα μπορούσε να ξεχάσει ή να βαρεθεί να ξεκινήσει χειροκίνητα μία

καταγραφή. Σκοπός μας είναι να αποτυπώσουμε το ποσοστό άνθρακα που εκπέμπει με όσο το δυνατόν μεγαλύτερη ακρίβεια, όχι να υπολογίσουμε ένα μέρος αυτού.

Έτσι, έπρεπε να δημιουργηθεί μια λύση η οποία θα διόρθωνε τα τυχόν λάθη και τους πιθανούς κινδύνους που παρουσιάστηκαν. Η λύση, ήταν να εκτιμάμε αυτόματα τη μετακίνηση που πραγματοποιεί ο χρήστης, χωρίς να χρειάζεται να τη δηλώνει αυτός χειροκίνητα κάθε φορά.

Συνεπώς, δημιουργήθηκε η ανάγκη για τη συλλογή δεδομένων των αισθητήρων, για κάθε διαφορετική μετακίνηση. Οι μετακινήσεις που θέλαμε να εξετάσουμε ήταν το περπάτημα, το τρέξιμο, η μετακίνηση με το ποδήλατο, το αυτοκίνητο, το λεωφορείο και το τρένο. Οι αισθητήρες που χρησιμοποιήθηκαν προκειμένου να μπορέσουμε να ξεχωρίσουμε και να κατατάξουμε τις διαφορετικές μετακινήσεις, ήταν το επιταχυνσιόμετρο, το γυροσκόπιο και μαγνητόμετρο. Επιπλέον, είχαμε ως σκοπό να μη συμπεριλάβουμε τον αισθητήρα του GPS, μιας και αυτό θα έκανε την κατανάλωση της μπαταρίας, σαφώς μεγαλύτερη. Αν η εφαρμογή δεν είναι λειτουργική από όλες τις απόψεις, όπως εμφάνιση, ευχρηστία και κατανάλωση μπαταρίας, τότε δεν πρόκειται να χρησιμοποιηθεί από τους χρήστες.

Για αυτό λοιπόν τον σκοπό, δημιουργήθηκε η πρώτη κινητή εφαρμογή, η οποία ονομάστηκε SensorsApp και διανεμήθηκε στους χρήστες που θέλησαν να βοηθήσουν. Η εφαρμογή αυτή κατέγραφε τις ενδείξεις των τριών αυτών αισθητήρων και τις αποθήκευε σε αρχεία. Ουσιαστικά, πριν ξεκινήσει ο χρήστης χειροκίνητα μια καταγραφή, επέλεγε το είδος της μετακίνησης που ήθελε να καταγράψει. Έτσι κάθε αρχείο, είχε ως όνομα τη συγκεκριμένη χρονική στιγμή που ξεκίνησε η καταγραφή της μετακίνησης και το όνομα της μετακίνησης αυτής. Όταν ένας χρήστης είχε συλλέξει καταγεγραμμένες δραστηριότητες, μπορούσε να τις ανεβάζει μέσω της ίδιας της εφαρμογής στο cloud. Μετά την ολοκλήρωση της διαδικασίας της συλλογής των καταγραφών, έγινε αξιολόγηση και αρχειοθέτηση τους. Το επόμενο στάδιο ήταν η δημιουργία ενός μοντέλου τεχνητής νοημοσύνης, βασισμένο στα υπάρχοντα δεδομένα, προκειμένου να πραγματοποιείται η αυτόματη αναγνώριση της μετακίνησης.

Συνεχίζοντας λοιπόν, έπρεπε να εξάγουμε από τα δεδομένα συγκεκριμένα χαρακτηριστικά, προκειμένου να μπορούμε να αναγνωρίζουμε και να ξεχωρίζουμε τις διαφορετικές μετακινήσεις που πραγματοποιεί ο χρήστης. Έτσι, εξάγαμε συγκεκριμένες στατιστικές μετρικές. Οι μετρικές αυτές, υπολογίσθηκαν για κάθε τιμή, κάθε αισθητήρα, στους τρεις άξονες x, y και z. Οι μετρικές αυτές ήταν η μέση τιμή, η τυπική απόκλιση, η ελάχιστη και η μέγιστη τιμή, το εύρος, η μέση τετραγωνική ρίζα της μέσης τιμής και το (signal magnitude area).

Υστερα, τα δεδομένα εισήχθησαν στο Weka, προκειμένου να αποφασιστεί ποιο μοντέλο θα χρησιμοποιηθεί. Οι αλγόριθμοι που εξετάστηκαν είχαν πολύ μεγάλο ποσοστό ακρίβειας στον διαχωρισμό των διαφορετικών μετακινήσεων. Έτσι,

εξήχθει ένα μοντέλο κανόνων το οποίο θα μπορούσε λαμβάνοντας ως είσοδο τις τιμές των αισθητήρων, να προβλέψει το είδος της μετακίνησης που πραγματοποιούταν. Τι μπορεί να σημαίνει όμως αυτό ουσιαστικά;

Σημαίνει ότι η δυνατότητα αυτόματης αναγνώρισης της μετακίνησης χρησιμοποιώντας μόνο το επιταχυνσιόμετρο, το γυροσκόπιο και μαγνητόμετρο είναι απόλυτα εφικτή. Έτσι, ξεκίνησε η υλοποίηση της δεύτερης και τελευταίας κινητής εφαρμογής. Η δεύτερη εφαρμογή ονομάστηκε RecognitionApp και σκοπός της ήταν να αξιοποιεί το προαναφερθέν μοντέλο. Ουσιαστικά η εφαρμογή, καταλάβαινε σε πραγματικό χρόνο ποια μετακίνηση πραγματοποιεί ο χρήστης και πρόσφερε τη δυνατότητα προβολής στατιστικών στοιχείων και αρκετών διαγραμμάτων, προκειμένου ο χρήστης να μπορεί να αντιληφθεί το είδος της μετακίνησης που πραγματοποιεί περισσότερο, είτε γενικά, είτε ανά καταγραφή. Επίσης, η εφαρμογή πρόσφερε τη δυνατότητα προβολής του αποτυπώματος άνθρακα για κάθε χρήστη, βασισμένο είτε ανά μήνα είτε ανά μέρα.

Τέλος, ένα μεγάλο πλεονέκτημα που προέκυψε από την αυτόματη αναγνώριση της μετακίνησης, ήταν ότι ο χρήστης μπορούσε να ξεκινήσει την αναγνώριση και να πραγματοποιεί οποιαδήποτε μετακίνηση επιθυμούσε, χωρίς να χρειάζεται να παρεμβάλλεται η εφαρμογή. Η εφαρμογή καταλάβαινε αυτόματα ποια μετακίνηση πραγματοποιούταν εκείνη τη στιγμή.

3.2 Τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν

Οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη και την υλοποίηση της εργασίας, δίνονται παρακάτω με λεπτομέρεια.

Αρχικά, έπρεπε να αποφασιστεί η τεχνολογία ανάπτυξης της πρώτης και της δεύτερης κινητής εφαρμογής. Λαμβάνοντας υπόψη τις υπάρχουσες τεχνολογίες κατά την περίοδο ανάπτυξης, αυτές οι οποίες μελετήθηκαν ήταν οι εξής. Ο κλασικός τρόπος ανάπτυξης Android εφαρμογών μέσω του Android Studio, το React Native και το Flutter. Το React Native, ουσιαστικά, παρέχει ένα wrapper σε JavaScript, το οποίο επιτρέπει να αναπτυχθούν native εφαρμογές για κινητά, χρησιμοποιώντας τη γλώσσα προγραμματισμού JavaScript. Δηλαδή, οι εφαρμογές που αναπτύσσονται με React Native, είναι σαν να έχουν αναπτυχθεί μέσω του Android Studio. Το Flutter, είναι ακριβώς το ίδιο με το React Native, με την παραμετροποίηση ότι αντί για JavaScript, για να αναπτύξει κάποιος εφαρμογές χρησιμοποιώντας αυτή την τεχνολογία, θα πρέπει να χρησιμοποιήσει τη γλώσσα προγραμματισμού Dart. Η τεχνολογία που χρησιμοποιήθηκε εν τέλει ήταν το React Native. Ο λόγος για τον οποίο επιλέχθηκε η συγκεκριμένη τεχνολογία, είναι η γλώσσα προγραμματισμού που χρησιμοποιεί και η προϋπάρχουσα εξοικείωση που υπήρχε με τη βιβλιοθήκη.

Στη συνέχεια, αφότου ολοκληρώθηκε η πρώτη φάση, δηλαδή, είχαμε συλλέξει τα δεδομένα, έπρεπε να τα αναλύσουμε. Για αυτόν τον σκοπό επιλέχθηκε η γλώσσα προγραμματισμού, Python. Πιο συγκεκριμένα, η Python, είναι ίσως η καλύτερη γλώσσα προγραμματισμού όσον αφορά τη διαχείριση και την οπτικοποίηση των δεδομένων. Επίσης, κάνει τη συγγραφή κώδικα πολύ εύκολη, λόγω της σύνταξης της και λόγω του τεράστιου αριθμού εξωτερικών βιβλιοθηκών που υπάρχουν διαθέσιμες. Για την οπτικοποίηση των δεδομένων, χρησιμοποιήθηκαν τα `matplotlib` notebooks. Ο λόγος πίσω από αυτή την απόφαση, είναι το γεγονός πως κάθε διάγραμμα που δημιουργείται και κάθε αποτέλεσμα μίας εντολής, παραμένει εμφανές στον χρήστη υπό τη μορφή ιστορικού. Έτσι, αν θέλαμε να εξετάσουμε αν μία καταγραφή ανταποκρινόταν στη μετακίνηση που δήλωσε ο χρήστης κατά την καταγραφή της, απλά έπρεπε να δημιουργήσουμε ένα διάγραμμα αυτής και να το συγκρίνουμε με τα διαγράμματα που δημιουργήθηκαν από τις καταγραφές της ίδιας δραστηριότητας άλλων χρηστών. Επόμενο βήμα, ήταν αφότου διορθώσαμε και κατατάξαμε όλες τις καταγραφές, να εξάγουμε από αυτές, όλες τις στατιστικές μετρικές που θα χρησιμοποιούσαν για τη δημιουργία του μοντέλου της τεχνητής νοημοσύνης. Για αυτόν λοιπόν τον λόγο χρησιμοποιήθηκαν δύο πολύ γνωστές βιβλιοθήκες μαθηματικών. Η `numpy` και η `scipy`. Χρησιμοποιώντας αυτές τις δύο βιβλιοθήκες, ήμασταν σε θέση να υπολογίσουμε τα χαρακτηριστικά που θα χρειαζόνταν, χωρίς να χρειαστεί να γράψουμε περιττό κώδικα. Συνεπώς, αφότου υπολογίσαμε τα χαρακτηριστικά, τα εξάγαμε σε ένα αρχείο προκειμένου να μπορέσουμε να τα εισάγουμε στο τελευταίο εργαλείο που χρησιμοποιήθηκε.

Το τελευταίο εργαλείο που χρησιμοποιήθηκε, ήταν το Weka. Το Weka είναι ένα εργαλείο το οποίο περιέχει αλγόριθμους μηχανικής μάθησης και επιτρέπει στον χρήστη να εκτελέσει τους αλγόριθμους αυτούς στα δεδομένα που εισάγει κάθε φορά. Έτσι λοιπόν, εισάγαμε στο Weka το αρχείο που εξάγαμε μέσω της Python. Μόλις αποφασίστηκε, ποίος αλγόριθμος θα χρησιμοποιηθεί με βάση τα αποτελέσματα του, πήραμε το μοντέλο που εξήγαγε και το εισάγαμε στη δεύτερη και τελευταία κινητή εφαρμογή.

3.3 Διαδικασία ανάπτυξης της εφαρμογής

3.3.1 Διαδικασία αυτόματης εκτίμησης του μέσου μετακίνησης μέσω της εφαρμογής

Το λογισμικό που αναπτύχθηκε για την αυτόματη εκτίμηση του μέσου της μετακίνησης ονομάστηκε `RecognitionApp` και ήταν μια κινητή εφαρμογή. Σκοπός της εφαρμογής αυτής, ήταν να αναγνωρίζει αυτόματα και σε πραγματικό χρόνο, το είδος της μετακίνησής του χρήστη που θα τη χρησιμοποιούσε. Προκειμένου να μπορέσει να πετύχει αυτό τον σκοπό, η εφαρμογή κατέγραφε τα δεδομένα τριών

αισθητήρων, δημιουργούσε κάποιες στατιστικές μετρικές και ύστερα τις μετρικές αυτές τις χρησιμοποιούσε για να εκτιμήσει το είδος της μετακίνησης. Οι αισθητήρες που χρησιμοποιήθηκαν ήταν το επιταχυνσιόμετρο, το γυροσκόπιο και το μαγνητόμετρο. Ο τρόπος με τον οποίο η εφαρμογή ήταν σε θέση να εκτιμά αυτόματα το είδος της μετακίνησης, οφείλεται στο μοντέλο τεχνητής νοημοσύνης που εξήχθει από τα δεδομένα. Ουσιαστικά, το μοντέλο αυτό ήταν ένα σύνολο κανόνων. Οι κανόνες αυτοί, δέχονταν ως είσοδο τις προαναφερθείσες στατιστικές μετρικές. Αν οι μετρικές ήταν εντός των ορίων των συνθηκών των κανόνων, δηλαδή, οι συνθήκες ελέγχου ήταν αληθείς, τότε το μοντέλο επέστρεφε το είδος της μετακίνησης. Ύστερα, το είδος της μετακίνησης που εκτιμήθηκε, αποθηκευόταν για τη δημιουργία στατιστικών. Προφανώς, η αυτόματη εκτίμηση μπορούσε να πραγματοποιηθεί, ανεξάρτητα με το αν ο χρήστης ήταν εντός της εφαρμογής ή όχι. Εξάλλου, αν δεν μπορούσε να λειτουργήσει όταν ο χρήστης δεν ήταν εντός της εφαρμογής ή όταν είχε την οθόνη του κινητού απενεργοποιημένη, τότε το λογισμικό θα ήταν εντελώς άχρηστο.

Το επόμενο κομμάτι της εφαρμογής ήταν η λειτουργία των στατιστικών. Κατά την αυτόματη εκτίμηση της μετακίνησης, το αποτέλεσμα που παραγόταν από το μοντέλο αποθηκευόταν. Ύστερα, το αποτέλεσμα αυτό, καθώς και τα ήδη υπάρχοντα, χρησιμοποιούνται για την παραγωγή στατιστικών γραφημάτων και για τον υπολογισμό του αποτυπώματος άνθρακα. Αν η εφαρμογή, δεν μπορούσε να αποθηκεύει τα αποτελέσματα του μοντέλου και να παράγει στατιστικά διαγράμματα, τότε ουσιαστικά, η χρήση της θα ήταν μηδενική. Σκοπός, ήταν η αποθήκευση και προβολή των δεδομένων μετακίνησης, όχι η στυγνή εκτίμηση.

Το σενάριο χρήσης της εφαρμογής έχει ως εξής. Ο χρήστης που επιθυμεί να καταγράψει τη μετακίνησή του, ανοίγει την εφαρμογή και ενεργοποιεί την αυτόματη εκτίμηση. Ύστερα, μπορεί είτε να παραμείνει εντός της εφαρμογής, είτε να εξέλθει από αυτή. Μπορεί ακόμα και να απενεργοποιήσει την οθόνη του κινητού του. Η εφαρμογή θα εξακολουθήσει να αναγνωρίζει και να αποθηκεύει το είδος της μετακίνησής του καθώς και να υπολογίζει το αποτύπωμα του άνθρακα που παράγει. Μόλις ο χρήστης επιθυμεί να σταματήσει την αυτόματη αναγνώριση, εισέρχεται εντός της εφαρμογής και τη διακόπτει.

Σε αυτό το σημείο, αξίζει να σημειωθεί το εξής. Η αποθήκευση και η δημιουργία στατιστικών γραφημάτων, δε σχετίζεται με το αν ο χρήστης βρίσκεται εντός της εφαρμογής ή όχι. Τα στατιστικά διαγράμματα, δημιουργούνται αυτόματα και σε πραγματικό χρόνο. Έτσι, αν κάποιος ενεργοποιούσε την αυτόματη εκτίμηση μετακίνησης και κατευθυνόταν στην οθόνη της εφαρμογής, όπου παρουσιαζόταν τα στατιστικά δεδομένα, θα παρατηρούσε πως τα γραφήματα καθώς και το αποτύπωμα άνθρακα του, θα δημιουργούταν εκείνη τη στιγμή σε πραγματικό χρόνο.

3.3.2 Προγραμματιστική ανάλυση

Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας την τεχνολογία React Native. Για να μπορέσει κάποιος να αναπτύξει εφαρμογές για κινητά τηλέφωνα χρησιμοποιώντας το React Native, θα πρέπει να γνωρίζει πως λειτουργεί το React. Το React είναι ουσιαστικά το ίδιο πράγμα με το React Native, με τη διαφορά ότι το πρώτο, χρησιμοποιείται για την ανάπτυξη εφαρμογών διαδικτύου. Οι μόνες διαφορές που εντοπίζονται, είναι στον τρόπο απεικόνισης δεδομένων στην οθόνη, π.χ αντί για τη χρήση του `<div>` που χρησιμοποιείται στις εφαρμογές διαδικτύου, στο κινητό χρησιμοποιούμε το `<View>` κλπ.

Έχοντας αναλύσει λοιπόν την τεχνολογία ανάπτυξης της εφαρμογής, ας δούμε με λεπτομέρεια τις προγραμματιστικές αποφάσεις που πάρθηκαν. Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι εξής. Για την εμφάνιση της εφαρμογής, χρησιμοποιήθηκε η βιβλιοθήκη `react-native-kitten`. Πρόκειται για μια βιβλιοθήκη, η οποία παρέχει διάδραση με τον χρήστη και προσφέρει έτοιμα γραφικά στοιχεία, όπως κουμπιά, κάρτες κτλ. Η βιβλιοθήκη αυτή παρέχει και την επιλογή για την εναλλαγή του θέματος εμφάνισης. Οι δύο επιλογές που παρέχει είναι το φωτεινό και το σκούρο θέμα. Ο χρήστης μπορεί να επιλέξει όποιο θέμα επιθυμεί μέσα από τις ρυθμίσεις της εφαρμογής. Επομένη βιβλιοθήκη ήταν το `React Navigation`, η οποία δίνει τη δυνατότητα δημιουργίας οθονών και τη δυνατότητα περιήγησης εντός αυτών. Συνολικά, υπήρχαν πέντε οθόνες εντός της εφαρμογής. Η αρχική οθόνη ενεργοποίησης και απενεργοποίησης της αυτόματης εκτίμησης της μετακίνησης, η οθόνη προβολής των γενικών στατιστικών, η οθόνη προβολής των στατιστικών ενός συγκεκριμένου μήνα, καθώς και η οθόνη προβολής στατιστικών μιας συγκεκριμένης μέρας. Όλες οι οθόνες στατιστικών, περιλάμβαναν ένα διάγραμμα για την οπτική απεικόνιση των αποθηκευμένων δεδομένων. Τα διαγράμματα ήταν ραβδοδιαγράμματα και διαγράμματα γραμμών. Για τη δημιουργία των γραφημάτων, χρησιμοποιήθηκε η βιβλιοθήκη `react-native-svg-charts`. Τελευταία οθόνη, ήταν οι ρυθμίσεις, όπου ο χρήστης μπορούσε να αλλάξει το θέμα εμφάνισης της εφαρμογής. Η επόμενη βιβλιοθήκη που χρησιμοποιήθηκε ήταν η `react-native-sensors`, για την καταγραφή του επιταχυνσιόμετρου, του γυροσκοπίου και του μαγνητόμετρου. Η βιβλιοθήκη αυτή, επιστρέφει τα δεδομένα αυτών των αισθητήρων σε μια διαρκή βάση. Συνεπώς, για να μπορέσει να καταγράψει κάποιος τα αποτελέσματα των αισθητήρων, αρκεί να κάνει μια συνδρομή (`subscribe`) σε αυτά και να αποθηκεύει τις τιμές που επιστρέφονται.

Για την αποθήκευση των δεδομένων, χρησιμοποιήθηκαν δυο βιβλιοθήκες. Η `react-native-async-storage` και η `Realm`. Όσον αφορά, η πρώτη πρόκειται για μια βάση δεδομένων ζευγαριών κλειδιού-τιμής (`key-value pairs`). Η δεύτερη ήταν μια σχεσιακή βάση δεδομένων. Ο λόγος για τον οποίο χρησιμοποιήθηκαν δυο βάσεις δεδομένων, ήταν απλός. Προκειμένου να μπορέσουμε να αποθηκεύσουμε το θέμα εμφάνισης που θα επέλεγε ο χρήστης, θα έπρεπε να το αποθηκεύουμε και να το

διαβάζουμε κατά τη στιγμή εκκίνησης της εφαρμογής, ωστόσο, η τιμή που έπρεπε να διαβάζουμε από τη μονάδα αποθήκευσης ήταν μια. Το θέμα εμφάνισης που έχει επιλέξει ο χρήστης αν ήταν φωτεινό ή σκούρο. Συνεπώς, αν αυτή τη μια τιμή, την εισάγαμε στη σχεσιακή βάση, δηλαδή στη Realm, θα έπρεπε να φτιάξουμε ένα καινούριο πίνακα όπου θα περιείχε μια μόνο τιμή. Επιπλέον, αν κάποια στιγμή στο μέλλον θέλαμε να εισάγουμε άλλη μια τιμή ίδιου τύπου, θα έπρεπε να ακολουθήσουμε την ίδια διαδικασία, δηλαδή, να φτιάξουμε ένα καινούριο πίνακα στη βάση δεδομένων και να του εισάγουμε την τιμή αυτή. Κάτι τέτοιο, θα σήμαινε ότι πρέπει να αλλάξουμε το σχήμα της βάσης, άρα είτε να διαγράψουμε τα παλιά δεδομένα, πράγμα απευκταίο, ή να μεταφέρουμε το παλιό σχήμα στο καινούριο, κάτι το οποίο είναι εξαιρετικά ακριβό υπολογιστικά. Αναλύοντας περαιτέρω τη σχεσιακή βάση, η επιλογή της έγινε με κριτήριο το είδος των δεδομένων που θέλαμε να αποθηκεύσουμε καθώς και με βάση τον όγκο τους. Υπήρχε στη βάση δεδομένων, ένας μόνο πίνακας με το όνομα “Activity” (Δραστηριότητα) και ως πεδία, είχε τον τύπο της μετακίνησης (type), τη χρονιά και τον μήνα (yearAndMonth), την ημέρα (day), καθώς και τη χρονοσφραγίδα ανίχνευσης του συγκεκριμένου τύπου μετακίνησης (timestamp). Η τελευταία βιβλιοθήκη που χρησιμοποιήθηκε ονομαζόταν react-native-foreground-service και χρησίμευε για τη διατήρηση της αυτόματης εκτίμησης στο παρασκήνιο. Η βιβλιοθήκη αυτή, επέτρεπε τη διατήρηση της ανίχνευσης στο παρασκήνιο, ακόμα και κατά τη διάρκεια όπου η οθόνη του κινητού ήταν απενεργοποιημένη, δηλώνοντας τη διαδικασία αυτή στο λειτουργικό σύστημα, ως μια υπηρεσία που πρέπει να παραμείνει ενεργή. Το λειτουργικό σύστημα, δέχονταν αυτή την οδηγία και συνεπώς, διατηρούσε ασταμάτητη τη διαδικασία, καθ’ όλη τη διάρκεια λειτουργίας της.

Σε αυτό το σημείο, αξίζει να αναφερθεί ότι τα διαγράμματα και γενικότερα, όλη η διαδικασία υπολογισμού και εμφάνισης των στατιστικών δεδομένων, συνέβαινε σε πραγματικό χρόνο. Για παράδειγμα, αν ο χρήστης ενεργοποιούσε την αυτόματη εκτίμηση και κατευθυνόταν στις οθόνες των στατιστικών, θα έβλεπε τα διαγράμματα να δημιουργούνται και να ενημερώνονται εκείνη ακριβώς τη στιγμή. Όσον αφορά τα αποθηκευμένα δεδομένα, δηλαδή τον τύπο της μετακίνησης που εκτιμήθηκε, η δυνατότητα διαχείρισής τους από τον χρήστη ήταν περιορισμένη. Αυτό συνέβη για τον εξής λόγο. Αν μια μετακίνηση εκτιμόνταν λάθος από το μοντέλο και η εφαρμογή έδινε τη δυνατότητά στον χρήστη να την αλλάξει, τότε η εφαρμογή θα έχανε τελείως την αξιοπιστία της και κατά συνέπεια, το αποτύπωμα άνθρακα που θα εμφάνιζε θα ήταν λανθασμένο. Συνεπώς, για να υπάρχει διαφάνεια, η μόνη ενέργεια που μπορούσε να πραγματοποιήσει ο χρήστης στα αποθηκευμένα δεδομένα ήταν η διαγραφή τους.

3.3.3 Προβλήματα που παρουσιάστηκαν και πώς επιλύθηκαν

Το μοναδικό πρόβλημα που προέκυψε κατά την ανάπτυξη της εφαρμογής, ήταν ο τρόπος αποθήκευσης των δεδομένων της αυτόματης εκτίμησης με τρόπο

τέτοιο, έτσι ώστε να μπορούν να γίνουν ερωτήματα σε αυτά, στη βάση δεδομένων. Ο τρόπος με τον οποίο η Realm δέχεται τα ερωτήματά, είναι εντελώς διαφορετικός από την τυπική γλώσσα ερωτημάτων SQL. Πιο συγκεκριμένα, το πρόβλημα εστιάζονταν στον διαχωρισμό της χρονιάς και του μήνα, από την ημέρα. Θέλαμε να εξετάζουμε πόσοι διαφορετικοί μήνες υπάρχουν συνολικά στη βάση δεδομένων. Αυτό σήμαινε, ότι έπρεπε να διαχωρίσουμε τη χρονιά και τον μήνα από την ημέρα, και να βρούμε τις μοναδικές τιμές τους. Όμως, η ημερομηνία, αρχικά αποθηκευόταν ολόκληρη με τη μορφή χρονιά-μήνας-μέρα. Η σύνταξη που χρησιμοποιεί η Realm, δεν επιτρέπει την τροποποίηση ενός πεδίου του πίνακα κατά την εμφάνιση των αποτελεσμάτων. Έτσι δεν ήταν δυνατόν να διαλέξουμε τους μήνες μιας χρονιάς μοναδικά, αφού με αυτό τον τρόπο, θα διαλέγαμε και τις μοναδικές μέρες του μήνα αυτού. Η λύση ήταν εν' τέλη, να αποθηκεύεται ξεχωριστά η ημερομηνία, δηλαδή, αντί με την κλασική μορφή, χρονιά-μήνας-μέρα, αποθηκευόταν ως, χρονιά-μήνας και ξεχωριστά η ημέρα. Έτσι, όταν κάναμε την ερώτηση στη βάση δεδομένων να επιστρέφει τις μοναδικές τιμές χρονιάς και μήνα, επιστρέφονταν οι μήνες για τους οποίους υπήρχαν εγγραφές, μια φορά ο κάθε ένας. Αν για παράδειγμα είχαμε τις εγγραφές, 2020-04, 2020-04 και 2020-05, η απάντηση που επιστρέφονταν ήταν, 2020-04 και 2020-05.

3.4 Σύγκριση της εφαρμογής με αντίστοιχες

Αντίστοιχη εφαρμογή με την παρούσα, μπορεί να θεωρηθεί η εφαρμογή “Capture”[30]. Η εφαρμογή αυτή έχει ως σκοπό, μεταξύ άλλων, να καταγράψει το αποτύπωμα άνθρακα των χρηστών χρησιμοποιώντας τον αισθητήρα του GPS καθώς και τη χειροκίνητη καταχώρησή εγγραφής μετακίνησης από τον χρήστη. Ουσιαστικά, θέλει να παρουσιάσει στον χρήστη το συνολικό αποτύπωμα άνθρακα του και τους τρόπους με τους οποίους μπορεί να το μειώσει. Η ημερομηνία κυκλοφορίας αυτής της εφαρμογής ήταν 20 Ιουνίου του 2020. Γνωρίζοντας λοιπόν την αντίστοιχη εφαρμογή, ας τη συγκρίνουμε με την παρούσα.

Η παρούσα εφαρμογή όπως έχει ήδη αναφερθεί, ως σκοπό, έχει την αυτόματη καταγραφή του μέσου μετακίνησης των χρηστών με απώτερο σκοπό, την καταγραφή του αποτυπώματος άνθρακα τους. Ο τρόπος με τον οποίο αυτό συμβαίνει είναι μέσω των αισθητήρων κίνησης και προσανατολισμού. Συνεπώς, η μέθοδος αναγνώρισης είναι διαφορετική. Το πλεονέκτημα που απορρέει από την παρούσα εφαρμογή, όσον αφορά τη μέθοδο αναγνώρισης, είναι πως είμαστε σε θέση να αντιλαμβανόμαστε πλήρως το μέσο της μετακίνησης μιας και βασιζόμαστε στις κινήσεις του ανθρώπινου σώματος κατά τη διάρκεια των μετακινήσεων αυτών. Όμως, η παρούσα μέθοδος έχει και ένα μειονέκτημα. Η καταγραφή του αποτυπώματος άνθρακα δεν μπορεί να είναι τόσο ακριβής μιας και βασίζεται στη διάρκεια εκτέλεσης των μετακινήσεων και όχι στην απόσταση και την ταχύτητά του μέσου.

Επιπλέον, η ανάπτυξη της παρούσας εφαρμογής ξεκίνησε τον Οκτώβρη του 2019, σχεδόν έναν χρόνο πριν τη συγκρινόμενη. Αυτό υποδηλώνει πως η ιδέα και η ανάπτυξη μιας τέτοιας εφαρμογής για τον σκοπό αυτό, είναι ένα θέμα που αποτελεί ενδιαφέρον τόσο ερευνητικό όσο και εμπορικό.

4. Η Εφαρμογή για τη Συλλογή των Δεδομένων

Σε αυτή την ενότητα θα εξεταστεί η μεθοδολογία ανάπτυξης της κινητής εφαρμογής για τη συλλογή των δεδομένων των χρηστών. Πιο συγκεκριμένα θα αναλυθεί το σκεπτικό ανάπτυξης, οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν, η διαδικασία συγγραφής του πηγαίου κώδικα, η προγραμματιστική ανάλυσή του, καθώς και τα προβλήματα που παρουσιάστηκαν και πως επιλύθηκαν.

4.1. Ανάπτυξη της εφαρμογής για τη συλλογή δεδομένων

Η κινητή εφαρμογή που αναπτύχθηκε για την συλλογή δεδομένων των χρηστών, ονομάστηκε SensorsApp. Ο σκοπός της εφαρμογής, ήταν να καταγράφει τις ενδείξεις των αισθητήρων του κινητού κατά τη διάρκεια μιας συγκεκριμένης μετακίνησης και να τις αποθηκεύει στη συσκευή. Οι αισθητήρες που χρησιμοποιήθηκαν ήταν το επιταχυνσιόμετρο, το γυροσκόπιο και το μαγνητόμετρο. Κάλλιστα κάποιος θα μπορούσε να αναρωτηθεί γιατί δεν επιλέξαμε να συμπεριλάβουμε και τον αισθητήρα GPS. Η επιλογή αυτή έγινε σκόπιμα, μιας και δε θέλαμε η εφαρμογή να έχει μεγάλη κατανάλωση μπαταρίας. Επίσης, θέλαμε να εξετάσουμε αν είναι εφικτό να πραγματοποιηθεί η αυτόματη εκτίμηση μιας μετακίνησης βασιζόμενοι στη σχετική κίνηση του ατόμου, δηλαδή, στην κίνηση που έχει το ανθρώπινο σώμα κατά τη διάρκεια μιας μετακίνησης. Έτσι λοιπόν, ξεκίνησε η ανάπτυξη της εφαρμογής. Η τεχνολογία που χρησιμοποιήθηκε ήταν το React Native.

Το σενάριο χρήσης της εφαρμογής έχει ως εξής. Ο χρήστης που επιθυμεί να καταγράψει μια μετακίνηση του, ανοίγει την εφαρμογή και επιλέγει μιας από τις διαθέσιμες μετακινήσεις. Οι διαθέσιμες μετακινήσεις είναι το περπάτημα, το τρέξιμο, η μετακίνηση χρησιμοποιώντας το ποδήλατο, το αστικό λεωφορείο, το αυτοκίνητο και το τρένο. Έπειτα, ξεκινάει την καταγραφή. Η εφαρμογή αναπτύχθηκε με το σκεπτικό, ότι όσο ο χρήστης θα έκανε τη μετακίνηση αυτή, δε θα αλληλεπιδρούσε με την εφαρμογή ή θα αλληλεπιδρούσε με μια άλλη, π.χ κατά τη μετακίνηση με το αστικό. Αυτή η απόφαση πάρθηκε με δεδομένο ότι η βιβλιοθήκη που χρησιμοποιήθηκε για τη διατήρηση της εφαρμογής στο παρασκήνιο, δεν επέτρεπε την ταυτόχρονη καταγραφή και αλληλεπίδραση του χρήστη. Κατά τη διάρκεια ανάπτυξης της δεύτερης εφαρμογής, έγινε αντιληπτό πως κάτι τέτοιο δε θα ήταν δυνατόν να συμβεί και συνεπώς, διορθώθηκε. Τα δεδομένα που συλλέγονται στο παρασκήνιο από τους αισθητήρες του κινητού, αποθηκεύονται σε

μοναδικά αρχεία, με το όνομα τους να είναι ανάλογο με τη χρονική στιγμή έναρξης της καταγραφής. Όταν ο χρήστης ολοκληρώσει μια μετακίνηση, ανοίγει την εφαρμογή και σταμάτα την καταγραφή.

Προκειμένου να μπορέσουμε να συλλέξουμε τις καταγραφές και αυτό να γίνει με όσο το δυνατόν πιο εύκολο τρόπο για τον τελικό χρήστη, δημιουργήθηκε η δυνατότητα ανεβάσματος όλων των αποθηκευμένων καταγραφών σε μια υπηρεσία διαμοιρασμού αρχείων. Πιο συγκεκριμένα, η υπηρεσία που επιλέχθηκε ήταν το Dropbox. Η επιλογή αυτή έγινε με κριτήριο, την ευκολία ένταξης του API (Application Programming Interface) που παρείχε το Dropbox. Αξίζει να αναφερθεί η δέουσα προσοχή που δόθηκε όσον αφορά την ιδιωτικότητα των χρηστών. Προκειμένου να μην είναι δυνατόν να στοχοποιηθεί ένας χρήστης με βάση τον αριθμό και το είδος των μετακινήσεών του, η εφαρμογή δημιουργούσε ένα μοναδικό αναγνωριστικό αριθμό για τον κάθε χρήστη και χρησιμοποιούσε αυτό τον αριθμό για να αρχειοθετεί τις μετακινήσεις του στην υπηρεσία διαμοιρασμού. Κατά αυτό τον τρόπο, όλοι οι χρήστες παραμένουν ανώνυμοι.

4.1.1 Προγραμματιστική ανάλυση

Ας δούμε αναλυτικά τις προγραμματιστικές αποφάσεις που πάρθηκαν για την ανάπτυξη της εφαρμογής. Στην αρχή, η εφαρμογή αναπτύχθηκε χρησιμοποιώντας τις κλάσεις που παρέχει το React. Κατά τη διάρκεια ανάπτυξης όμως, το React κυκλοφόρησε τα Hooks. Τα Hooks, παρέχουν τις μεθόδους που χρησιμοποιούν οι κλάσεις, με τη διαφορά ότι είναι φτιαγμένα για να χρησιμοποιούνται σε τμηματικά κομμάτια. Για παράδειγμα, αντί να πρέπει να δημιουργηθεί μια ολόκληρη κλάση για τη δημιουργία μιας οθόνης, δημιουργείται ένα τμηματικό κομμάτι. Το πλεονέκτημα που παρέχει ένα τέτοιο τμηματικό κομμάτι, είναι πως μπορεί να παρέχει την ίδια λειτουργικότητα, απλά με πολύ λιγότερο boilerplate κώδικα. Συνεχίζοντας, οι βιβλιοθήκες που χρησιμοποιήθηκαν σε όλα τα μέρη της εφαρμογής, αναλύονται παρακάτω. Ως βιβλιοθήκη αλληλεπίδρασης χρήστη, επιλέχθηκε το react-native-paper. Η βιβλιοθήκη αυτή παρέχει ότι περιλαμβάνουν τα κριτήρια σχεδίασης του υλικού σχεδιασμού (material design) της Google, στις εφαρμογές που έχουν αναπτυχθεί χρησιμοποιώντας το React Native. Επιπλέον, ανάμεσα στο λευκό και το σκούρο θέμα εμφάνισης, η εφαρμογή χρησιμοποιούσε το σκούρο. Για την περιήγηση του χρήστη στην εφαρμογή, χρησιμοποιήθηκε η βιβλιοθήκη React Navigation. Συνολικά, υπήρχαν δυο οθόνες για την αλληλεπίδραση του χρήστη με την εφαρμογή. Η οθόνη επιλογής των διαθέσιμων μετακινήσεων προς καταγραφή και η οθόνη ανεβάσματος των αποθηκευμένων καταγραφών στην υπηρεσία διαμοιρασμού αρχείων. Για την καταγραφή του επιταχυνσιόμετρου, του γυροσκοπίου και του μαγνητόμετρου χρησιμοποιήθηκε η βιβλιοθήκη react-native-sensors. Η βιβλιοθήκη αυτή, επιστρέφει τα δεδομένα αυτών των αισθητήρων σε μια διαρκή βάση. Συνεπώς, για να μπορέσει να καταγράψει κάποιος τα αποτελέσματα των αισθητήρων, αρκεί να κάνει μια συνδρομή (subscribe) σε αυτά

και να αποθηκεύει τις τιμές που επιστρέφουν. Για την αποθήκευση των τιμών σε αρχεία στο χώρο αποθήκευσης του τηλεφώνου, καθώς και για το ανέβασμα τους στην υπηρεσία διαμοιρασμού, χρησιμοποιήθηκε η βιβλιοθήκη `react-native-fetch-blob`. Η βιβλιοθήκη αυτή, παρέχει τη δυνατότητα αποθήκευσης δεδομένων σε αρχεία στον χώρο αποθήκευσης του κινητού, καθώς και τη δυνατότητα αποστολής και λήψης αρχείων από το διαδίκτυο. Για την καταγραφή των αισθητήρων στο παρασκήνιο, χρησιμοποιήθηκε η βιβλιοθήκη `react-native-background-job`.

Σε αυτό το σημείο, αξίζει να αναρωτηθεί κανείς, γιατί τα δεδομένα των αισθητήρων δεν αποθηκευόταν σε μια βάση δεδομένων αλλά σε αρχεία στον εσωτερικό χώρο αποθήκευσης του κινητού. Ο τρόπος αποθήκευσης των δεδομένων έγινε σε αρχεία για την καλύτερη ταξινόμηση και διαχείρισή τους. Αν τα αρχεία ήταν αποθηκευμένα σε μια βάση δεδομένων, δε θα ήταν δυνατόν να διαχωριστούν με τρόπο χωρικά βέλτιστο. Αν για παράδειγμα, υποθέσουμε ότι ως μοναδικό κλειδί χρησιμοποιούσαμε τη χρονική στιγμή έναρξης μιας καταγραφής, τότε για όλες τις τιμές, όλων των αισθητήρων, θα χρειαζόταν να κρατάμε την ίδια πληροφορία πάρα πολλές φορές. Πιο συγκεκριμένα, κάθε αισθητήρας επιστρέφει 5 τιμές. Το πρώτο γραμμάτιο του αισθητήρα που χρησιμοποιήθηκε, το X, το Y, το Z και τη χρονοσφραγίδα εμφάνισης των συγκεκριμένων τιμών. Αν ως μοναδικό αναγνωριστικό κλειδί, χρησιμοποιούσαμε τη χρονική στιγμή εκκίνησης μιας εγγραφής, τότε αν είχαμε δέκα χιλιάδες τιμές αισθητήρων για αυτή την καταγραφή, θα είχαμε επιπλέον δέκα χιλιάδες ίδιες τιμές για τη χρονική στιγμή εκκίνησης. Το ίδιο θα συνέβαινε και με το είδος της μετακίνησης. Παρότι το είδος της μετακίνησης θα ήταν το ίδιο, θα έπρεπε να υπάρχει για κάθε εγγραφή στη βάση. Για αυτούς λοιπόν τους λόγους χρησιμοποιήθηκαν αρχεία για την αποθήκευση των καταγραφών. Με αυτό τον τρόπο, κάθε καταγραφή χωριζόταν σε διαφορετικά αρχεία και ως όνομα κάθε αρχείου, χρησιμοποιούσαν τη χρονική στιγμή εκκίνησης της καταγραφής μαζί με τη μετακίνηση που πραγματοποιούσαν. Έτσι, για το ανέβασμα και επίσης για την ανάλυση των καταγραφών μεταγενέστερα, απλώς, αποστέλλονταν αυτό το αρχείο στην υπηρεσία διαμοιρασμού. Ύστερα, για την ανάλυση αυτής της μετακίνησης, δημιουργούσαν ένα διάγραμμα με όλο το περιεχόμενο του αρχείου. Με αυτό τον τρόπο όλη η αποθήκευση και η διαχείρισή στον χώρο διαμοιρασμού γινόταν με έμμεσο τρόπο, αυτόματα.

Συνεχίζοντας, η επομένη προγραμματιστική επιλογή που πραγματοποιήθηκε, ήταν η απόφαση για τον τρόπο αποστολής και τον χώρο αποθήκευσης των αρχείων στο διαδίκτυο (cloud). Έπρεπε να αποφασιστεί ο τρόπος με τον οποίο τα αρχεία θα αποστέλλονταν από τα κινητά των χρηστών, σε μια υπηρεσία διαμοιρασμού, προκειμένου να μπορέσουμε να τα συλλέξουμε και στη συνέχεια να τα αναλύσουμε. Εξετάστηκαν αρκετές επιλογές πάροχων υπηρεσιών αποθήκευσης, ωστόσο η πιο απλή στην ενσωμάτωση, ήταν η υπηρεσία διαμοιρασμού αρχείων, Dropbox. Εκτός από τις υπηρεσίες διαμοιρασμού αρχείων, έγινε και ανάλυση για τη δημιουργία ενός διακομιστή και ενός REST API. Ωστόσο, μια τέτοια λύση θα ήταν χρονοβόρα και στην

προκειμένη περίπτωση άσκοπη. Από τη στιγμή που υπάρχει μια υπηρεσία η οποία είναι πάντα διαθέσιμη και έχει δοκιμαστεί από εκατομμύρια χρήστες, δεν υπάρχει λόγος να μην την προτιμήσουμε. Σκοπός μας, ήταν να αποστείλουμε όλα τα αρχεία καταγραφών που υπήρχαν στο κινητό ενός χρήστη, στον φάκελο με το αναγνωριστικό κλειδί του, στην υπηρεσία διαμοιρασμού, με όσο το δυνατόν λιγότερο περίπλοκο τρόπο. Έτσι, η διαδικασία είχε ως εξής. Προκειμένου να μπορέσουμε να αποστείλουμε τα αρχεία στην υπηρεσία, χωρίς όμως να χρησιμοποιήσουμε κάποια δική της εφαρμογή, έπρεπε να δημιουργηθεί ένα μοναδικό κλειδί API στον λογαριασμό υποδοχής τους. Αν τα αρχεία δεν ανέβαιναν στον ίδιο λογαριασμό, τότε θα έπρεπε οι χρήστες να αποστέλουν χειροκίνητα τον σύνδεσμο για την πρόσβαση σε αυτά, από τον δικό τους λογαριασμό Dropbox. Κάτι τέτοιο, προφανώς, δε θα αξιοποιούσε την υπηρεσία για τον σκοπό που επιλέχθηκε. Μια τέτοια λύση θα ήταν ίδια και με το να αναγκάζαμε τους χρήστες να βρουν μόνοι τους τα αρχεία στη συσκευή τους και να τα στείλουν χειροκίνητα μέσω email. Έτσι, κάθε χρήστης που θα εγκαθιστούσε την εφαρμογή, αν ήθελε να ανεβάσει τα αρχεία καταγραφών του, απλά θα κατευθυνόταν στην αντίστοιχη οθόνη και θα πατούσε το κουμπί ανεβάσματος. Κατά την ολοκλήρωση του ανεβάσματος, τα αρχεία διαγράφονταν από τη συσκευή του χρήστη για σκοπούς καθαρισμού της συσκευής και για να μην υποβάλουμε τον χρήστη στη διαδικασία να τα ψάξει για να τα σβήσει ο ίδιος. Σκοπός της εφαρμογής, ήταν να προσφέρει όσο το δυνατόν καλύτερη εμπειρία χρήσης. Δηλαδή να μη χρειάζεται να σκέφτεται ο χρήστης, απλά να αντιδρά.

Όπως αναλύθηκε και παραπάνω, εκτός από το να συλλέξουμε δεδομένα, σκοπός μας ήταν και να σεβαστούμε την ιδιωτικότητα των χρηστών. Αν ξέραμε ποιος χρήστης έχει κάνει ποιες και πόσες μετακινήσεις, πολύ εύκολα θα προσβαλλόταν το δικαίωμα του στην ιδιωτικότητα. Συνεπώς, κατά την εγκατάσταση της εφαρμογής, αυτόματα δημιουργούταν ένα τυχαίο μοναδικό αναγνωριστικό κλειδί. Αυτό το κλειδί, ήταν ουσιαστικά το όνομα του χρήστη στην υπηρεσία διαμοιρασμού των αρχείων. Όλες οι καταγραφές που θα πραγματοποιούσε αυτός ο χρήστης, θα αποστέλλονταν μέσα σε αυτό τον φάκελο. Ένα άλλο πλεονέκτημα που απορρέει από αυτό τον τρόπο αποθήκευσης, είναι ότι μπορούσαμε να γνωρίζουμε τον ακριβή αριθμό χρηστών και τον ακριβή αριθμό καταγραφών ανά χρήστη. Δε χρειαζόταν να κατεβάσουμε τα δεδομένα και να τρέξουμε κάποιο είδος ερωτήματος (query) σε αυτά.

4.1.2 Προβλήματα που παρουσιάστηκαν και πώς επιλύθηκαν

Τα προβλήματα που παρουσιάστηκαν κατά την ανάπτυξη της πρώτης κινητής εφαρμογής, επικεντρωθήκαν κυρίως στη διαδικασία της καταγραφής των δεδομένων από τους αισθητήρες, όσο η εφαρμογή ήταν στο παρασκήνιο. Αν η εφαρμογή, δεν μπορούσε να λειτουργήσει όταν ο χρήστης έκλεινε την οθόνη του κινητού του ή όταν χρησιμοποιούσε κάποια άλλη εφαρμογή, τότε δε θα ήταν

λειτουργική. Συνεπώς, σκοπός της εφαρμογής ήταν να τρέχει στο παρασκήνιο, να συλλέγει και να αποθηκεύει τα δεδομένα των αισθητήρων σε αρχεία.

Η συμπεριφορά που έχει μια απλή android εφαρμογή είναι η ακόλουθη. Όσο η εφαρμογή είναι ανοιχτή στο προσκήνιο, δηλαδή, στην οθόνη του χρήστη, τότε λειτουργεί κανονικά. Όταν η εφαρμογή μπει στο παρασκήνιο, δηλαδή ο χρήστης βγει από την οθόνη της, αυτή, συνεχίζει την οποία διεργασία μπορεί να εκτελούσε όσο ήταν στο προσκήνιο για κάποια δευτερόλεπτα και μετά, αυτόματα, το λειτουργικό σύστημα τη σταματάει. Αυτό, συμβαίνει διότι μια εφαρμογή, δεν μπορεί να συνεχίσει να εκτελείται επ' αόριστον στο παρασκήνιο, να καταναλώνει πολύτιμους πόρους και κατά συνέπεια σπαταλάει μπαταρία. Έτσι λοιπόν, αν μια εφαρμογή επιθυμεί να εκτελέσει μια διεργασία στο παρασκήνιο και η διεργασία αυτή, ξεκινάει με την αλληλεπίδραση του χρήστη, τότε πρέπει να ενημερώσει το λειτουργικό σύστημα ότι πρέπει να συνεχιστεί και στο παρασκήνιο. Στο android, αυτή η οδηγία προς το λειτουργικό σύστημά ονομάζεται *service* και εκχωρείται στο λειτουργικό σύστημα κατά τη στιγμή της εκτέλεσης. Αναλύοντας περαιτέρω το λειτουργικό σύστημα του android από μια σκοπιά υψηλού επιπέδου, εκτός από τα *services* που μπορεί να εκμεταλλευτεί μια εφαρμογή, υπάρχουν και τα *tasks*. Ένα *task*, είναι ουσιαστικά μια διαδικασία η οποία εκτελείται στο παρασκήνιο, χωρίς όμως να απαιτείται κάποια αλληλεπίδραση με τον χρήστη. Κάτι τέτοιο είναι χρήσιμο στις περιπτώσεις που μια εφαρμογή χρειάζεται να ανανεώσει το περιεχόμενο της στο παρασκήνιο, έτσι ώστε όταν ο χρήστης την ανοίξει ξανά, να μη χρειαστεί να περιμένει να φορτώσει το περιεχόμενο αυτό. Βέβαια, η ανανέωση αυτή στο παρασκήνιο, δεν μπορεί να πραγματοποιηθεί οποιαδήποτε στιγμή. Έτσι, αν μια εφαρμογή θέλει να τρέξει ένα *task*, ενημερώνει το λειτουργικό σύστημα και αυτό με τη σειρά του, το τοποθετεί σε μια ουρά. Ο χρόνος εκτέλεσης από την ουρά είναι άγνωστος μιας και αυτό εξαρτάται καθαρά από το λειτουργικό σύστημα. Το μόνο που μπορεί να εγγυηθεί το λειτουργικό σύστημα, είναι ότι το *task* αυτό θα εκτελεστεί και δε θα λιμοκτονήσει.

Ωστόσο, κατά τη διάρκεια ανάπτυξης της εφαρμογής, αυτή την πληροφορία δεν τη γνωρίζαμε. Συνεπώς, για να επιτευχθεί αυτή η διαδικασία, δηλαδή να είναι σε θέση η εφαρμογή να συνεχίσει την καταγραφή στο παρασκήνιο, χρειάστηκε να πραγματοποιηθεί μια εναλλακτική λύση. Διαβάζοντας τα παραπάνω, αντιλαμβανόμαστε πως το *task* και το *service*, είναι δύο τελείως διαφορετικά πράγματα, όπου το κάθε ένα, χρησιμοποιείται για τελείως διαφορετικούς σκοπούς. Κατά συνέπεια, ένα *task* δεν μπορεί να αντικαταστήσει τον ρόλο ενός *service* και αντίστροφα. Όμως, η εναλλακτική λύση που αναφέρθηκε παραπάνω, κάνει ακριβώς αυτό.

Έτσι, όταν ο χρήστης επιλέξει το είδος της μετακίνησής που θέλει να πραγματοποιήσει και ξεκινήσει τη διαδικασία της καταγραφής, τότε ενεργοποιείται ένα *task* στο παρασκήνιο. Όμως, επειδή ένα *task* δε γίνεται να εκτελεστεί όσο η εφαρμογή είναι στο προσκήνιο, ο χρήστης πρέπει να εξέλθει της εφαρμογής.

Αφότου εξέλθει και μετά το πέρασμα 5 δευτερολέπτων, η εφαρμογή εισάγει στην ουρά του λειτουργικού συστήματος τη διεργασία για την παρακολούθηση των αισθητήρων. Όμως, στη δική μας περίπτωση, δε γινόταν να περιμένουμε πότε το λειτουργικό σύστημα θα αποφάσιζε να εκτελέσει το task. Το task έπρεπε να εκτελεστεί αμέσως. Για αυτό λοιπόν τον σκοπό, χρησιμοποιήθηκε μια μέθοδος η οποία προορίζεται για δοκιμαστική χρήση και μόνο. Πιο συγκεκριμένα, όταν ένας προγραμματιστής θέλει να διαπιστώσει αν ένα task πρόκειται να εκτελεστεί σωστά, δηλαδή αν ο κώδικας που έχει γραφτεί είναι σωστός και επιτυγχάνει τον σκοπό του, τότε μπορεί να αναγκάσει το λειτουργικό σύστημα να το εκτελέσει αμέσως χωρίς να χρειαστεί να περιμένει στην ουρά. Αυτό ακριβώς επιτυγχάνει και η δική μας λύση. Κάθε 5 δευτερόλεπτα, δημιουργείται ένα καινούριο task και κάθε task, έχει διάρκεια εκτέλεσης 5 δευτερόλεπτα. Έτσι, δημιουργείται ένας ατελείωτος βρόγχος επαναλήψεων δημιουργίας και τερματισμού tasks, τα οποία εκτελούνται αμέσως. Και κατά αυτό τον τρόπο λοιπόν, ήμασταν σε θέση, να συνεχίζουμε την εκτέλεση της καταγραφής και της αποθήκευσης των ενδείξεων των αισθητήρων, για όσο χρονικό διάστημα επιθυμούσε ο χρήστης. Όταν ο χρήστης ολοκλήρωνε μια μετακίνηση, τότε άνοιγε πάλι την εφαρμογή και σταματούσε την εν λόγω διαδικασία. Όσο απρόσμενο και αν φαίνεται, στην πράξη λειτούργησε με εξαιρετική ακρίβεια και μας εξασφάλισε τα δεδομένα που χρειαζόσασταν για τη δημιουργία του μοντέλου μηχανικής μάθησης.

4.1.3 Διαδικασία δοκιμής του λογισμικού

Μόλις ολοκληρώθηκε η πρώτη εφαρμογή και δοκιμάστηκε, έπρεπε να δοθεί στους χρήστες, προκειμένου να συλλεχθούν τα δεδομένα των διαφορετικών μετακινήσεων. Το γεγονός πως επιλέχθηκε η εφαρμογή να χρησιμοποιηθεί από πολλούς και διαφορετικούς χρήστες δεν έγινε για εμπορικούς σκοπούς ή για σκοπούς καθαρής δοκιμής του λογισμικού. Από τη στιγμή που σκοπός μας ήταν να συλλέξουμε δεδομένα μετακινήσεων, έπρεπε να έχουμε και πρόσβαση για την πραγματοποίηση των μετακινήσεων αυτών. Ένας άλλος παράγοντας, ήταν διαφορετικότητα των δεδομένων. Αν το μοντέλο τεχνητής νοημοσύνης εκπαιδευόταν με μόνο μια πηγή αλήθειας, δηλαδή τις καταγραφές από μόνο ένα άτομο, τότε δε θα ήταν και αποτελεσματικό. Για παράδειγμα, ο τρόπος βαδίσματος διαφέρει από άνθρωπο σε άνθρωπο, ο τρόπος τρεξίματος, ο τρόπος με τον οποίο κινείται το ανθρώπινο σώμα μέσα στο αστικό λεωφορείο, στο αυτοκίνητο ή στο τρένο.

Συνεπώς, το λογισμικό διανεμήθηκε στους χρήστες. Οι χρήστες ήταν φοιτητές του τμήματος πληροφορικής του Ιονίου Πανεπιστημίου. Συνολικά σε αριθμό ήταν 13, συμπεριλαμβανομένου και του προγραμματιστή της εφαρμογής.

5. Τα Δεδομένα και το Μοντέλο για την Αυτόματη Εκτίμηση του Μέσου Μετακίνησης

Σε αυτή την ενότητα θα εξεταστεί η διαδικασία ανάλυσης και εξόρυξης πληροφορίας, για τα δεδομένα που συλλέχθηκαν. Πιο συγκεκριμένα, θα αναλυθεί η διαδικασία αξιολόγησης των δεδομένων, η διαδικασία επιδιόρθωσης τους, η διαδικασία υπολογισμού των στατιστικών μετρικών που χρειάστηκαν για την υλοποίηση του μοντέλου τεχνητής νοημοσύνης, η διαδικασία εξαγωγής του αρχείου στο σύνολο δεδομένων (dataset) και τέλος, η διαδικασία εύρεσης του μοντέλου καθώς και η διαδικασία παραγωγής των κανόνων για την κινητή εφαρμογή αυτόματης εκτίμησης του μέσου μετακίνησης.

5.1 Αξιολόγηση των δεδομένων

Μόλις ολοκληρώθηκε η διαδικασία συλλογής των δεδομένων, συγκεντρώθηκαν τα δεδομένα από την υπηρεσία διαμοιρασμού αρχείων Dropbox και άρχισε η διαδικασία ανάλυσής τους. Σε αυτή τη φάση, έπρεπε να εξεταστεί αν τα δεδομένα των χρηστών ήταν αληθή, δηλαδή, αν τα δεδομένα που έστειλαν οι χρήστες ανταποκρίνονταν στις μετακινήσεις που είχαν δηλώσει κατά την καταγραφή τους. Για να επιτευχθεί όμως αυτό, έπρεπε να υπάρχει μια σαφής εικόνα για την κάθε μετακίνηση, έτσι ώστε όταν θα γινόταν η σύγκριση μεταξύ των μετακινήσεων των χρηστών, να μπορούσε να αξιολογηθεί αν η εκάστοτε καταγραφή ήταν σωστή. Σωστή θεωρούνταν μια καταγραφή, όταν τα δεδομένα που περιείχε ήταν του τύπου μετακίνησης που δηλώθηκε κατά την καταγραφή της. Για αυτόν λοιπόν τον σκοπό, όλες οι κατηγορίες μετακινήσεων καταγράφηκαν μια φορά από τον προγραμματιστή της εφαρμογής. Αυτό έγινε με το εξής σκεπτικό. Ο προγραμματιστής της εφαρμογής, ως δημιουργός της, θα ακολουθήσει ρητά τις οδηγίες που πρέπει να ακολουθήσει ένας χρήστης προκειμένου να πραγματοποιήσει μια καταγραφή σωστά. Αν για παράδειγμα ήθελε να καταγράψει δεδομένα μετακίνησης με το αστικό λεωφορείο, τότε τα δεδομένα που θα είχε συλλέξει στο τέλος της μετακίνησης, θα ήταν από αυτό το είδος μετακίνησης και μόνο. Κατά αυτό τον τρόπο, υπήρχε μια βάση αλήθειας, δηλαδή, καταγραφές οι οποίες αντιπροσώπευαν σωστά το κάθε είδος μετακίνησης. Αυτό, έγινε διότι ένα πιθανό σενάριο λάθους ήταν, ο χρήστης ο οποίος θα κατέγραφε ένα συγκεκριμένο είδος μιας μετακίνησης, να ξέχναγε ότι καταγράφει και να άλλαζε το είδος της μετακίνησής του με αποτέλεσμα, αντί για παράδειγμα να κατέγραφε μετακίνηση με το αστικό λεωφορείο, να κατέγραφε δεδομένα αστικού λεωφορείου και μετακίνησης με τα πόδια, στο ίδιο αρχείο καταγραφής. Συνεπώς, όταν θα έπρεπε αυτά τα δεδομένα να αναλυθούν και να συγκριθούν με άλλα δεδομένα ίδιου

τύπου μετακίνησης, τα αποτελέσματα δε θα έμοιαζαν, συνεπώς η εκπαίδευση του μοντέλου τεχνητής νοημοσύνης θα ήταν λανθασμένη και κατ' επέκταση, η αυτόματη εκτίμηση θα υπολόγιζε λανθασμένα το μέσο των μετακινήσεων.

5.2. Επιδιόρθωση των δεδομένων

Επόμενο βήμα ήταν να εξεταστεί αν όλα τα αρχεία καταγραφών ανταποκρινόταν στο είδος της μετακίνησης που είχε δηλωθεί κατά την καταγραφή τους. Για αυτόν λοιπόν τον σκοπό, όλα τα δεδομένα οπτικοποιήθηκαν σε διαγράμματα και ομαδοποιήθηκαν ανά είδος μετακίνησης. Έπειτα, για την κάθε ομαδοποιημένη κατηγορία και για το κάθε αρχείο μέσα σε αυτή, γινόταν οπτική σύγκριση με τα διαγράμματα των αρχείων της βάσης αλήθειας. Ακολουθώντας τη συγκεκριμένη διαδικασία ελέγχου, μπορέσαμε να επιβεβαιώσουμε αν οι καταγραφές ανταποκρινόταν στο είδος της μετακίνησής που είχε δηλώσει ο χρήστης κατά την καταγραφή τους. Ως εργαλείο για τον παραπάνω έλεγχο, χρησιμοποιήθηκαν τα jupyter notebooks, η γλώσσα προγραμματισμού Python και η βιβλιοθήκη matplotlib.

5.3 Υπολογισμός των στατιστικών μετρικών

Αφότου διορθώθηκαν τυχόν λάθη στις καταγραφές των χρηστών, ξεκίνησε η διαδικασία δημιουργίας στατιστικών μετρικών για την αναγνώριση των διαφορετικών μετακινήσεων. Σε αυτό το σημείο βέβαια, αξίζει να αναρωτηθεί κάποιος, για ποιο λόγο έπρεπε να υπολογίσουμε συγκεκριμένες στατιστικές μετρικές αντί απλά να συγκρίνουμε τα αποτελέσματα των αισθητήρων μεταξύ τους. Η απάντηση, αν και αρκετά απλή, δεν είναι προφανής, διότι, κάθε ίδια μετακίνηση δεν είναι ταυτόσημη με μια άλλη. Ναι, είναι αρκετά παρόμοιες αλλά δεν είναι ούτε ίδιες, ούτε ίσες. Για παράδειγμα, μια μετακίνηση με το αστικό λεωφορείο από τον χρήστη Α είναι εξαιρετικά όμοια με μια μετακίνηση ίδιου τύπου από τον χρήστη Β, αλλά, αν προσπαθήσουμε να συγκρίνουμε αυτές τις μετακινήσεις, το αποτέλεσμα δε θα είναι το ίδιο και αυτό, διότι εξαρτάται από πολλούς παράγοντες, όπως από την ταχύτητα του λεωφορείου, την επιτάχυνση του, τη διαδρομή που πραγματοποιεί, τη θέση του κινητού τηλεφώνου σε σχέση με το αστικό λεωφορείο και από τη σχετική κίνηση του σώματος του κάθε χρήστη σε σχέση τόσο με το λεωφορείο, όσο και με το κινητό τηλέφωνο. Ακόμα και ο χρήστης Α να πραγματοποιήσει την ίδια διαδρομή δυο φορές, στην ίδια θέση του λεωφορείου, με το λεωφορείο να πηγαίνει με την ίδια ταχύτητα και να έχει την ίδια ακριβώς επιτάχυνση, δε θα είναι ίδια τα νούμερα. Συνεπώς, με το να βασιστούμε σε αυτή τη σύγκριση για να εκτιμούμε αυτόματα το μέσο μετακίνησης, θα ήταν αποτυχία. Έτσι, αν από το κάθε είδος μετακίνησης και για την κάθε καταγραφή, εξάγαμε ένα σύνολο από στατιστικές μετρικές, τότε, θα μπορούσαμε να συγκρίνουμε αυτές τις μετρικές και σύμφωνα με τα αποτελέσματά

τους, να εξάγουμε συμπεράσματα για το είδος της μετακίνησης. Θα επεκταθούμε περισσότερο σε αυτό το κομμάτι στη συνέχεια, όπου θα γίνει λεπτομερής ανάλυση του μοντέλου και των κανόνων για την αυτόματη εκτίμηση του μέσου μετακίνησης που εξήχθησαν.

Συνεχίζοντας λοιπόν, ας εξετάσουμε αναλυτικά τις στατιστικές μετρικές που χρησιμοποιήθηκαν. Συνολικά, οι μετρικές ήταν 25 και υπολογίστηκαν για τις τιμές του κάθε αισθητήρα. Συνεπώς, συνολικά εξήχθησαν **25 (Στατιστικές μετρικές ανά αισθητήρα) x 3 (Το σύνολο των αισθητήρων) = 75 μετρικές**. Επόμενο βήμα ήταν ο υπολογισμός του αριθμού των τιμών που θα είχε κάθε παράθυρο υπολογισμού. Παράθυρο υπολογισμού ορίζεται ως ο χρόνος, εντός του οποίου, υπολογίζουμε τις 75 αυτές μετρικές. Η κάθε καταγραφή χωρίζεται σε χρονικά παράθυρα των 4 δευτερολέπτων. Για να μπορέσουμε να αντιληφθούμε το είδος της κίνησης που θα πραγματοποιούσε ένας χρήστης, θα έπρεπε να περιμένουμε ένα χρονικό διάστημα προκειμένου, να μεν να πραγματοποιηθεί η κίνηση αυτή, αλλά και να μπορέσουμε να υπολογίσουμε και τις μετέπειτα αλλαγές της κίνησης έγκαιρα. Δηλαδή, τη μετάβαση από το περπάτημα στη μετακίνηση με το αστικό λεωφορείο. Η επιλογή του νούμερου 4, δεν ήταν τυχαία. Ύστερα από έρευνες που έχουν πραγματοποιηθεί, έχει αποδειχθεί ότι η δραστηριότητα του ανθρώπινου σώματος κυμαίνεται από 1.5Hz μέχρι 15Hz. Αυτό μεταφράζεται από 1.5 μέχρι 15 αναγνώσεις των αισθητήρων κίνησής ανά δευτερόλεπτο. Τα 4 δευτερόλεπτα είναι αρκετός χρόνος για να πραγματοποιηθεί και να υπολογιστεί ένα είδος μετακίνησής. Στη δική μας περίπτωση, η εφαρμογή έκανε τη συλλογή δεδομένων κάθε 2.5Hz. Αυτό σημαίνει ότι κάθε δευτερόλεπτο, η εφαρμογή, διάβαζε 2.5 τιμές από τους αισθητήρες του επιταχυνσιόμετρου, του γυροσκοπίου και του μαγνητόμετρου. Για να μπορέσουμε λοιπόν να έχουμε ένα χρονικό παράθυρο παρατήρησης των τιμών των 4 δευτερολέπτων, έπρεπε να έχουμε 10 τιμές των αισθητήρων. Αναλυτικά, **2.5 (Τιμές ανά δευτερόλεπτο) x 4 (Δευτερόλεπτα) = 10 τιμές** ανά χρονικό παράθυρο. Έτσι ο υπολογισμός των στατιστικών μετρικών έγινε χρησιμοποιώντας 10 τιμές κάθε άξονα, κάθε αισθητήρα για κάθε χρονικό παράθυρο.

Αναλύοντας λοιπόν περαιτέρω τις στατιστικές μετρικές που χρησιμοποιήθηκαν, ακολουθεί η αναλυτική περιγραφή τους. Η πρώτη μετρική ήταν η **Μέση Τιμή**, δεύτερη η **Τυπική Απόκλιση**, τρίτη και τέταρτη ήταν η **Μέγιστη Τιμή** και η **Ελάχιστη Τιμή** αντίστοιχα. Πέμπτη τιμή ήταν το **Εύρος**, έκτη τιμή ήταν η **Τετραγωνική Ρίζα της Μέσης Τιμής** και έβδομη το **Εμβαδόν του Μεγέθους του Σήματος** ή αλλιώς, το **SMA (Signal Magnitude Area)**. Για όλες τις μετρικές εκτός της τελευταίας, υπολογίστηκε και το **SMV (Signal Magnitude Vector)** ή αλλιώς, το **Διάνυσμα μεγέθους του Σήματος**. Επισημαίνεται ξανά, ότι οι 8 αυτές μετρικές υπολογίστηκαν για κάθε άξονα.

Έτσι λοιπόν, ο υπολογισμός της **Μέσης Τιμής** έγινε για κάθε άξονα, κάθε αισθητήρα. Συνοπτικά, **Μέσος Όρος** ή αλλιώς **Δειγματική Μέση Τιμή** ενός συνόλου n

παρατηρήσεων αποτελεί το σπουδαιότερο και χρησιμότερο μέτρο της Στατιστικής και είναι ένα μέτρο θέσης, δηλαδή δείχνει σχετικά τις θέσεις των αριθμών στους οποίους αναφέρεται. Η μέση τιμή συμμετέχει σε αρκετούς τύπους της στατιστικής και εξετάζεται σε σχεδόν όλες τις στατιστικές κατανομές. Γενικά, ορίζεται ως το άθροισμα των παρατηρήσεων δια του πλήθους αυτών. Είναι δηλαδή η μαθηματική πράξη ανεύρεσης της «μέσης απόστασης» ανάμεσα σε δύο ή περισσότερους αριθμούς. Η μέση τιμή συμβολίζεται με $\bar{x}$. Ο γενικός τύπος της μέσης τιμής είναι:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n t_i = \frac{1}{n}(t_1 + \dots + t_n), \text{ όπου } t_i \text{ η } i \text{ παρατήρηση και } n \text{ το πλήθος των}$$

παρατηρήσεων.

Το SMV ορίζεται ως η ένταση του τριαξονικού σήματος. Ο γενικός τύπος του SMV είναι:

$SMV[t] = \sqrt{x[t]^2 + y[t]^2 + z[t]^2}$, όπου $x[t]$, $y[t]$ και $z[t]$ είναι οι 10 τιμές του X, του Y και του Z άξονα αντίστοιχα, στο χρόνο δειγματοληψίας t.

Ο τύπος που χρησιμοποιήθηκε για να υπολογιστεί το SMV της μέσης τιμής είναι:

$SMV[t] = \sqrt{\bar{x} \left(\sum_{t=1}^{10} x^2[t] \right) + \bar{y} \left(\sum_{t=1}^{10} y^2[t] \right) + \bar{z} \left(\sum_{t=1}^{10} z^2[t] \right)}$, όπου $\bar{x} \left(\sum_{t=1}^{10} x^2[t] \right)$, $\bar{y} \left(\sum_{t=1}^{10} y^2[t] \right)$ και $\bar{z} \left(\sum_{t=1}^{10} z^2[t] \right)$ είναι η μέση τιμή των 10 τιμών του X, του Y και του Z άξονα αντίστοιχα, στο χρόνο δειγματοληψίας t.

Ο υπολογισμός της **Μέσης Τιμής** και του **SMV της Μέσης Τιμής** με τη χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*.

```
# save the mean values of the last 10 axis values
mean_x = np.mean(ten_values_of_x)
mean_y = np.mean(ten_values_of_y)
mean_z = np.mean(ten_values_of_z)
mean_smv = np.sqrt(np.mean(np.power(ten_values_of_x, 2)) \
+ np.mean(np.power(ten_values_of_y, 2)) \
+ np.mean(np.power(ten_values_of_z, 2)))
```

Η επόμενη μετρική που υπολογίστηκε ήταν η **Τυπική Απόκλιση** για κάθε άξονα, κάθε αισθητήρα. Τυπική απόκλιση, είναι ένα μέτρο που χρησιμοποιείται για να υπολογιστεί το ποσό της μεταβολής ή της διασποράς ενός συνόλου τιμών δεδομένων. Μια χαμηλή τυπική απόκλιση υποδηλώνει ότι τα σημεία των δεδομένων

τείνουν να είναι κοντά στο μέσο όρο (που ονομάζεται επίσης η αναμενόμενη τιμή) του συνόλου, ενώ μία υψηλή τυπική απόκλιση υποδεικνύει ότι τα στοιχεία απλώνονται πάνω από ένα ευρύτερο φάσμα των τιμών. Ο γενικός τύπος της τυπικής απόκλισης είναι:

$$\sigma(x) = \sqrt{\frac{N}{N-1}} \sqrt{E[(x - E(x))^2]}$$

Ο τύπος που χρησιμοποιήθηκε για να υπολογιστεί το SMV της τυπικής απόκλισης είναι:

$$SMV[t] = \sqrt{\sigma\left(\sum_{t=1}^{10} x^2[t]\right) + \sigma\left(\sum_{t=1}^{10} y^2[t]\right) + \sigma\left(\sum_{t=1}^{10} z^2[t]\right)}, \text{ όπου } \sigma\left(\sum_{t=1}^{10} x^2[t]\right), \sigma\left(\sum_{t=1}^{10} y^2[t]\right) \text{ και } \sigma\left(\sum_{t=1}^{10} z^2[t]\right) \text{ είναι η τυπική απόκλιση των 10 τιμών του } X, \text{ του } Y \text{ και του } Z \text{ άξονα αντίστοιχα, στο χρόνο δειγματοληψίας } t.$$

Ο υπολογισμός της **Τυπικής Απόκλισης** και του **SMV της Τυπικής Απόκλισης** με την χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*.

```
# save the std values of the last 10 axis values
std_x = np.std(ten_values_of_x)
std_y = np.std(ten_values_of_y)
std_z = np.std(ten_values_of_z)
std_smv = np.sqrt(np.std(np.power(ten_values_of_x, 2)) \
    + np.std(np.power(ten_values_of_y, 2)) \
    + np.std(np.power(ten_values_of_z, 2)))
```

Επόμενη μετρική ήταν η **Μέγιστη Τιμή** για κάθε άξονα, κάθε αισθητήρα. Συνεπώς, για τον υπολογισμό της, επιλέγονταν η μέγιστη τιμή ανάμεσα στις 10, για το εκάστοτε χρονικό παράθυρο.

Ο τύπος που χρησιμοποιήθηκε για να υπολογιστεί το SMV της μέγιστης τιμής είναι:

$$SMV[t] = \sqrt{\max(x^2[t]) + \max(y^2[t]) + \max(z^2[t])}, \text{ όπου } \max(x^2[t]), \max(y^2[t]) \text{ και } \max(z^2[t]) \text{ είναι η μέγιστη τιμή των 10 τιμών του } X, \text{ του } Y \text{ και του } Z \text{ άξονα αντίστοιχα, στο χρόνο δειγματοληψίας } t.$$

Ο υπολογισμός της **Μέγιστη Τιμής** και του **SMV της Μέγιστης Τιμής** με τη χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*.

```
# save the max values of the last 10 axis values
```

```

max_x = np.max(ten_values_of_x)
max_y = np.max(ten_values_of_y)
max_z = np.max(ten_values_of_z)
max_smv = np.sqrt(np.max(np.power(ten_values_of_x, 2)) \
    + np.max(np.power(ten_values_of_y, 2)) \
    + np.max(np.power(ten_values_of_z, 2)))

```

Με ακριβώς τον ίδιο τρόπο υπολογίστηκε και η **Ελάχιστη Τιμή** για κάθε άξονα, κάθε αισθητήρα.

Ο τύπος που χρησιμοποιήθηκε για να υπολογιστεί το SMV της ελάχιστης τιμής είναι:

$SMV[t] = \sqrt{\min(x^2[t]) + \min(y^2[t]) + \min(z^2[t])}$, όπου $\min(x^2[t])$, $\min(y^2[t])$ και $\min(z^2[t])$ είναι η ελάχιστη τιμή των 10 τιμών του X, του Y και του Z άξονα αντίστοιχα, στο χρόνο δειγματοληψίας t.

Ο υπολογισμός της **Ελάχιστης Τιμής** και του **SMV της Ελάχιστης Τιμής** με την χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*.

```

# save the min values of the last 10 axis values
min_x = np.min(ten_values_of_x)
min_y = np.min(ten_values_of_y)
min_z = np.min(ten_values_of_z)
min_smv = np.sqrt(np.min(np.power(ten_values_of_x, 2)) \
    + np.min(np.power(ten_values_of_y, 2)) \
    + np.min(np.power(ten_values_of_z, 2)))

```

Επόμενη μετρική που υπολογίστηκε για κάθε άξονα, κάθε αισθητήρα ήταν το **Εύρος**. Ως εύρος, ορίζεται η διαφορά μεταξύ της μεγαλύτερης και της μικρότερης τιμής ενός συνόλου δεδομένων. Εκφράζεται με τον τύπο:

$$R(x) = \max(x) - \min(x)$$

Ο τύπος που χρησιμοποιήθηκε για να υπολογιστεί το SMV του εύρους είναι:

$SMV[t] = \sqrt{R(x^2[t]) + R(y^2[t]) + R(z^2[t])}$, όπου $R(x^2[t])$, $R(y^2[t])$ και $R(z^2[t])$ είναι το εύρος των 10 τιμών του X, του Y και του Z άξονα αντίστοιχα, στο χρόνο δειγματοληψίας t.

Ο Υπολογισμός του **Εύρους** και του **SMV του Εύρους** με τη χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*.

```
# save the range values of the last 10 axis values
range_x = np.ptp(ten_values_of_x)
range_y = np.ptp(ten_values_of_y)
range_z = np.ptp(ten_values_of_z)
range_smv = np.sqrt(np.ptp(np.power(ten_values_of_x, 2)) \
    + np.ptp(np.power(ten_values_of_y, 2)) \
    + np.ptp(np.power(ten_values_of_z, 2)))
```

Φτάνοντας προς το τέλος της ανάλυσης του υπολογισμού των μετρικών, επόμενη ήταν η **Τετραγωνική Ρίζα της Μέσης Τιμής** ή αλλιώς **Root Mean Square (RMS)**. Η τιμή **RMS** ενός συνόλου τιμών (ή μιας κυματομορφής συνεχούς χρόνου) είναι η τετραγωνική ρίζα του αριθμητικού μέσου όρου των τετραγώνων των τιμών ή το τετράγωνο της συνάρτησης που καθορίζει τη συνεχή κυματομορφή. Ο γενικός τύπος της τετραγωνικής ρίζας της μέσης τιμής είναι:

$$RMS(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n t_i^2} = \sqrt{\frac{1}{n}(t_1^2 + \dots + t_n^2)}, \text{ όπου } t_i \text{ η } i \text{ παρατήρηση και } n \text{ το}$$

πλήθος των παρατηρήσεων.

Ο τύπος που χρησιμοποιήθηκε για να υπολογιστεί το SMV της τετραγωνικής ρίζας της μέσης τιμής είναι:

$$SMV[t] = \sqrt{RMS(x[t])^2 + RMS(y[t])^2 + RMS(z[t])^2}, \text{ όπου } RMS(x[t])^2, RMS(y[t])^2 \text{ και } RMS(z[t])^2 \text{ είναι η τετραγωνική ρίζα της μέσης τιμής των 10 τιμών του } X, \text{ του } Y \text{ και του } Z \text{ άξονα αντίστοιχα, στο χρόνο δειγματοληψίας } t.$$

Ο υπολογισμός της **Τετραγωνικής Ρίζας της Μέσης Τιμής** και του **SMV της Τετραγωνικής Ρίζας της Μέσης Τιμής** με τη χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*.

```
# save the rms values of the last 10 axis values
rms_x = np.sqrt(np.mean(np.power(ten_values_of_x, 2)))
rms_y = np.sqrt(np.mean(np.power(ten_values_of_y, 2)))
rms_z = np.sqrt(np.mean(np.power(ten_values_of_z, 2)))
rms_smv = np.sqrt(rms_x + rms_y + rms_z)
```

Τέλος, η τελευταία μετρική που χρησιμοποιήθηκε ήταν το **Εμβαδόν του Μεγέθους του Σήματος** ή αλλιώς **Signal Magnitude Area (SMA)**. Στα μαθηματικά, η περιοχή μεγέθους του σήματος, είναι ένα στατιστικό μέτρο του μεγέθους μιας ποικίλης ποσότητας και ορίζεται από τον τύπο:

$$\chi_{SMA} = \sum_{i=1}^n |x_i|, \text{ όπου } x_i \text{ η } i \text{ παρατήρηση και } n \text{ το πλήθος των παρατηρήσεων.}$$

Ο υπολογισμός του **Εμβαδού του Μεγέθους του Σήματος** με τη χρήση της γλώσσας προγραμματισμού *Python* και της βιβλιοθήκης *numpy*:

```
# save the sma values of the last 10 axis values
sma = (np.sum(np.absolute(np.array(ten_values_of_x))) \
      + np.sum(np.absolute(np.array(ten_values_of_y))) \
      + np.sum(np.absolute(np.array(ten_values_of_z)))) / 10
```

5.4. Εξαγωγή του αρχείου δεδομένων (Dataset)

Έχοντας λοιπόν ολοκληρώσει τη διαδικασία υπολογισμού των στατιστικών μετρικών, έπρεπε να εξαχθεί ένα αρχείο δεδομένων (dataset) το οποίο θα περιείχε τις μετρικές για κάθε καταγραφή, χωρισμένες ανά γραμμή, με βάση το χρονικό παράθυρο, προκειμένου να εισαχθεί σε κάποιο αλγόριθμο μηχανικής μάθησης. Πιο συγκεκριμένα, η δομή του τελικού αρχείου δεδομένων είχε ως εξής.

Συνολικά, το αρχείο περιείχε, 23.370 γραμμές. Κάθε γραμμή, ήταν ένα χρονικό παράθυρο ενός συγκεκριμένου τύπου μετακίνησης. Κάθε παράθυρο περιείχε 75 μετρικές καθώς και τον τύπο της μετακίνησης για το χρονικό αυτό περιθώριο, προφανώς, μια καταγραφή μεγαλύτερη των τεσσάρων δευτερολέπτων, χωριζόταν σε πολλές γραμμές εντός του αρχείου. Για παράδειγμα, μια καταγραφή 12 δευτερολέπτων, κατείχε 3 γραμμές εντός του αρχείου δεδομένων.

Ακολουθεί πολύ συνοπτικά η δομή του αρχείου δεδομένων.

```
mean_accel_x, mean_accel_y, mean_accel_z, ..., std_accel_x, std_accel_y, mean_gyro_x, ..., ..., activity
mean_accel_x, mean_accel_y, mean_accel_z, ..., std_accel_x, std_accel_y, mean_gyro_x, ..., ..., activity
mean_accel_x, mean_accel_y, mean_accel_z, ..., std_accel_x, std_accel_y, mean_gyro_x, ..., ..., activity
```

5.5. Εύρεση του μοντέλου μηχανικής μάθησης

5.5.1 Οι αλγόριθμοι που δοκιμάστηκαν και τα αποτελέσματά τους

Τελευταίο στάδιο για τη δημιουργία του μοντέλου αυτόματης εκτίμησης, ήταν η επιλογή του αλγορίθμου μηχανικής μάθησης, μέσω του λογισμικού Weka[37]. Συνολικά, πραγματοποιήθηκαν 64 δοκιμές. Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ότι κατά τη διάρκεια των καταγραφών, υπήρχε σαν επιλογή καταγραφής και η μετακίνηση με τη μηχανή, ωστόσο, επειδή οι καταγραφές ήταν μόλις τέσσερεις, δε συμπεριλήφθηκαν καθόλου στη διαδικασία δημιουργίας του μοντέλου.

Συνεχίζοντας λοιπόν, οι 64 αυτές δοκιμές συμπεριλάμβαναν τους αλγόριθμους, J48[40], NaiveBayes[38], KNN[39] και τον αλγόριθμο PART[41].

Ο J48 είναι ένας αλγόριθμος που δημιουργεί ένα δέντρο αποφάσεων. Έτσι, αν αυτό το δέντρο προσπελαστεί, θα επιστρέψει και μια απόφαση. Οι NaiveBayes και KNN είναι υπολογιστικοί αλγόριθμοι, αυτό σημαίνει ότι δεν υπάρχει κάποια δομή που μπορεί να προσπελαστεί προκειμένου να επιστραφεί ένα αποτέλεσμα. Προκειμένου να παραχθεί η απόφαση, πρέπει να γίνουν υπολογισμοί εκείνη τη στιγμή. Προφανώς, κάτι τέτοιο δε γινόταν να πραγματοποιηθεί μιας και το μοντέλο θα εκτελούταν εσωτερικά του τηλεφώνου. Αν για κάθε απόφαση του μέσου μετακίνησης έπρεπε να περιμένουμε να πραγματοποιηθεί ο υπολογισμός αυτός, το κόστος σε χρόνο και μπαταρία, θα ήταν τεράστιο, συνεπώς, παρά τη σχετικά μεγαλύτερη ακρίβεια τους, δεν επιλέχθηκαν. Τέλος, ο αλγόριθμος PART. Ο αλγόριθμος αυτός είναι ένας αλγόριθμος παραγωγής κανόνων. Πιο συγκεκριμένα, ο αλγόριθμος παράγει ένα ενδιάμεσο δέντρο J48 και επιλέγει τη διαδρομή από τη ρίζα στο “καλύτερο” φύλλο, έπειτα, αυτή η διαδρομή γίνεται ένας κανόνας. Συνεπώς, το αποτέλεσμα αυτού του αλγόριθμου είναι ένα σύνολο κανόνων.

Ακολουθούν ενδεικτικά αποτελέσματα για τον κάθε αλγόριθμο που εξετάστηκε καθώς και τα φίλτρα που εφαρμόστηκαν για την παραγωγή κάθε αποτελέσματος.

Ακολουθούν τα αποτελέσματα του αλγόριθμου PART με επαλήθευση τύπου 10 fold cross validation.

<u>PART</u>							
Unbalanced				Balanced			
No Feature Selection		Feature Selection		No Feature Selection		Feature Selection	
<i>Number of Rules</i>	400	<i>Number of Rules</i>	446	<i>Number of Rules</i>	147	<i>Number of Rules</i>	151

Πίνακας 1. Αποτελέσματα αλγόριθμου PART για Unbalanced και Balanced

PART							
Unbalanced							
No Feature Selection				Feature Selection			
Accuracy	88.96 %			Accuracy	88.69 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.93	0.93	walking		0.92	0.93	walking
	0.87	0.86	car_riding		0.88	0.86	car_riding
	0.80	0.79	bicycling		0.79	0.78	bicycling
	0.84	0.83	train_riding		0.84	0.82	train_riding
	0.89	0.90	bus_riding		0.88	0.91	bus_riding
	0.88	0.89	running		0.90	0.88	running
Weighted Avg.	0.88	0.89		Weighted Avg.	0.88	0.88	

Πίνακας 2. Αναλυτικά αποτελέσματα αλγόριθμου PART για Unbalanced

PART							
Balanced							
No Feature Selection				Feature Selection			
Accuracy	82.06 %			Accuracy	82.14 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.80	0.79	walking		0.79	0.79	walking
	0.80	0.82	car_riding		0.78	0.82	car_riding
	0.83	0.81	bicycling		0.82	0.82	bicycling
	0.75	0.76	train_riding		0.79	0.74	train_riding

	0.78	0.78	<i>bus_riding</i>		0.79	0.79	<i>bus_riding</i>
	0.94	0.94	<i>running</i>		0.93	0.94	<i>running</i>
Weighted Avg.	0.82	0.82		Weighted Avg.	0.82	0.82	

Πίνακας 3. Αναλυτικά αποτελέσματα αλγόριθμου PART για *Balanced*

Ακολουθούν τα αποτελέσματα του αλγόριθμου J48 με επαλήθευση τύπου 10 fold cross validation.

J48							
Unbalanced				Balanced			
No Feature Selection		Feature Selection		No Feature Selection		Feature Selection	
<i>Number Of Leaves</i>	903	<i>Number Of Leaves</i>	964	<i>Number Of Leaves</i>	297	<i>Number Of Leaves</i>	293
<i>Size of the Tree</i>	1805	<i>Size of the Tree</i>	927	<i>Size of the Tree</i>	593	<i>Size of the Tree</i>	585

Πίνακας 4. Αποτελέσματα αλγόριθμου J48 για *Unbalanced* και *Balanced*

J48							
Unbalanced							
No Feature Selection				Feature Selection			
Accuracy	87.27%			Accuracy	87.16 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.92	0.93	<i>walking</i>		0.92	0.93	<i>walking</i>
	0.85	0.85	<i>car_riding</i>		0.86	0.84	<i>car_riding</i>
	0.78	0.76	<i>bicycling</i>		0.79	0.75	<i>bicycling</i>
	0.81	0.80	<i>train_riding</i>		0.80	0.79	<i>train_riding</i>

	0.87	0.87	bus_riding		0.87	0.88	bus_riding
	0.87	0.88	running		0.87	0.87	running
Weighted Avg.	0.87	0.87		Weighted Avg.	0.87	0.87	

Πίνακας 5. Αναλυτικά αποτελέσματα αλγόριθμου J48 για Unbalanced

J48							
Balanced							
No Feature Selection				Feature Selection			
Accuracy	80.33 %			Accuracy	80.58 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.76	0.76	walking		0.76	0.77	walking
	0.78	0.81	car_riding		0.78	0.79	car_riding
	0.80	0.80	bicycling		0.80	0.80	bicycling
	0.76	0.74	train_riding		0.76	0.75	train_riding
	0.76	0.75	bus_riding		0.78	0.77	bus_riding
	0.94	0.93	running		0.93	0.93	running
Weighted Avg.	0.80	0.80		Weighted Avg.	0.80	0.80	

Πίνακας 6. Αναλυτικά αποτελέσματα αλγόριθμου J48 για Balanced

Ακολουθούν τα αποτελέσματα του αλγόριθμου KNN (K Nearest Neighbours) με επαλήθευση τύπου 10 fold cross validation.

KNN							
Unbalanced							
No Feature Selection				Feature Selection			
Accuracy	91.66 %			Accuracy	90.08 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.95	0.95	walking		0.94	0.95	walking
	0.93	0.89	car_riding		0.90	0.86	car_riding
	0.89	0.89	bicycling		0.89	0.87	bicycling
	0.85	0.84	train_riding		0.85	0.82	train_riding
	0.90	0.92	bus_riding		0.87	0.91	bus_riding
	0.93	0.95	running		0.91	0.94	running
Weighted Avg.	0.91	0.91		Weighted Avg.	0.90	0.90	

Πίνακας 7. Αναλυτικά αποτελέσματα αλγόριθμου KNN για Unbalanced

KNN							
Balanced							
No Feature Selection				Feature Selection			
Accuracy	86.82 %			Accuracy	85.93 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.89	0.84	walking		0.88	0.83	walking
	0.91	0.83	car_riding		0.87	0.83	car_riding
	0.89	0.92	bicycling		0.88	0.91	bicycling

	0.79	0.77	train_riding		0.77	0.76	train_riding
	0.78	0.85	bus_riding		0.78	0.82	bus_riding
	0.94	0.98	running		0.95	0.78	running
Weighted Avg.	0.86	0.86		Weighted Avg.	0.85	0.85	

Πίνακας 8. Αναλυτικά αποτελέσματα αλγόριθμου KNN για *Balanced*

Ακολουθούν τα αποτελέσματα του αλγόριθμου Naive Bayes με επαλήθευση τύπου 10 fold cross validation.

Naive Bayes							
Unbalanced							
No Feature Selection				Feature Selection			
Accuracy	60.01 %			Accuracy	64.55 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.90	0.74	walking		0.90	0.75	walking
	0.50	0.57	car_riding		0.50	0.41	car_riding
	0.32	0.73	bicycling		0.34	0.71	bicycling
	0.41	0.70	train_riding		0.49	0.65	train_riding
	0.74	0.34	bus_riding		0.70	0.59	bus_riding
	0.72	0.89	running		0.70	0.89	running
Weighted Avg.	0.68	0.60		Weighted Avg.	0.68	0.64	

Πίνακας 9. Αναλυτικά αποτελέσματα αλγόριθμου Naive Bayes για *Unbalanced*

Naive Bayes							
Balanced							
No Feature Selection				Feature Selection			
Accuracy	68.08 %			Accuracy	67.49 %		
	Precision	Recall	Class		Precision	Recall	Class
	0.73	0.73	walking		0.74	0.73	walking
	0.61	0.53	car_riding		0.60	0.41	car_riding
	0.70	0.78	bicycling		0.70	0.77	bicycling
	0.52	0.66	train_riding		0.56	0.56	train_riding
	0.59	0.47	bus_riding		0.51	0.66	bus_riding
	0.93	0.90	running		0.93	0.90	running
Weighted Avg.	0.68	0.68		Weighted Avg.	0.67	0.67	

Πίνακας 10. Αναλυτικά αποτελέσματα αλγόριθμου Naive Bayes για Balanced

5.5.2 Η επιλογή του αλγόριθμου PART

Ο αλγόριθμος που επιλέχθηκε εν τέλει, ήταν το PART. Όμως, θα μπορούσε να αναρωτηθεί κάποιος, γιατί να επιλεγθεί ο PART έναντι του J48, αφού ουσιαστικά, πραγματοποιούν την ίδια διαδικασία. Η απάντηση βρίσκεται στο αποτέλεσμα του κάθε αλγορίθμου. Ναι, ο J48 επιστρέφει ένα δέντρο, ωστόσο, αυτό το δέντρο όταν είναι πολύ μεγάλο σε έκταση, είναι εξαιρετικά δύσκολο να μετατραπεί σε προγραμματιστικό κώδικά, εν αντιθέσει με το αποτέλεσμα του PART. Ο δεύτερος επιστρέφει ένα σύνολο κανόνων, δηλαδή, ένα σύνολο από προγραμματιστικά “Αν” (if) ή αλλιώς, συνθήκες ελέγχου. Έτσι η μετατροπή του σε κωδικά JavaScript, ήταν εξαιρετικά εύκολη. Ο τρόπος σύγκρισης των αποτελεσμάτων για τον κάθε αλγόριθμο είχε ως εξής. Αρχικά, ελέγχονταν τα αποτελέσματα χωρίς να γίνει κάποια επεξεργασία των δεδομένων. Ύστερα, γινόταν επιλογή χαρακτηριστικών χρησιμοποιώντας τον αλγόριθμο CfsSubsetEval και ο αλγόριθμος, δοκιμάζονταν ξανά. Επόμενο βήμα ήταν να ακολουθηθεί η παραπάνω διαδικασία ξανά, δηλαδή χρησιμοποιώντας τον ίδιο αλγόριθμο, άλλα με τη διαφορά, ότι πλέον τα δεδομένα

είχαν ισορροπηθεί (balanced) χρησιμοποιώντας τον αλγόριθμο του Weka, SpreadSubSample.

Κατά την ανάλυση των αποτελεσμάτων, αποδείχθηκε ότι η ακρίβειά των αποτελεσμάτων παρέμεινε σχεδόν η ίδια είτε εφαρμόζαμε επιλογή χαρακτηριστικών, είτε όχι. Συνεπώς, με δεδομένο ότι το μοντέλο θα εκτελούταν στην κινητή συσκευή, μας συνέφερε να μη χρειάζεται να υπολογίσουμε και τις 75 μετρικές για κάθε παράθυρο 4 δευτερολέπτων. Μετά την επιλογή των χαρακτηριστικών οι μετρικές μειώθηκαν στις 38, δηλαδή στις μισές, ωστόσο, παρατηρήθηκε ότι η ισορρόπηση των κλάσεων μείωνε την ακρίβεια των αποτελεσμάτων κατά ένα μεγάλο βαθμό, είτε εφαρμόζαμε επιλογή χαρακτηριστικών, είτε όχι, οπότε η ισορρόπηση δε χρησιμοποιήθηκε στην παράγωγή του τελικού μοντέλου.

5.6. Διαδικασία παραγωγής κανόνων για την κινητή εφαρμογή αυτόματης εκτίμησης του μέσου μετακίνησης

Έτσι λοιπόν, λαμβάνοντας υπόψιν όλα τα παραπάνω, ακολουθεί η διαδικασία με την οποία δημιουργήθηκε το μοντέλο αυτόματης εκτίμησης του μέσου μετακίνησης, μέσω του αλγορίθμου PART. Αρχικά, φορτώθηκε το αρχείο δεδομένων (dataset) στο Weka. Ύστερα, αφότου εφαρμόστηκε ο αλγόριθμος CfsSubsetEval για την επιλογή των χαρακτηριστικών, εκτελέστηκε ο αλγόριθμος PART. Έπειτα, αφότου πάρθηκαν τα αποτελέσματα, μετατράπηκαν σε προγραμματιστικό κώδικα JavaScript χρησιμοποιώντας τη γλώσσα προγραμματισμού Python. Έπειτα, φορτώθηκε το μοντέλο στον πηγαίο κώδικα της εφαρμογής RecognitionApp. Όταν ενεργοποιούνταν η διαδικασία αυτόματης εκτίμησης και περνούσε το χρονικό παράθυρο των τεσσάρων δευτερολέπτων, τα δεδομένα των αισθητήρων χρησιμοποιούνταν για να υπολογιστούν οι 38 στατιστικές μετρικές. Μέτα, οι μετρικές αυτές εισάγονταν στο μοντέλο και επιστρέφονταν σαν αποτέλεσμα, το μέσο μετακίνησης του χρήστη.

Ακολουθεί πολύ συνοπτικά η δομή των κανόνων που εξήχθησαν από το Weka καθώς και η δομή τους σε γλώσσα προγραμματισμού JavaScript.

- PART

```
std_accel_smv > 6.606645 AND  
std_accel_z <= 5.947899 AND  
sma_gyro > 0.562692 AND  
std_accel_y <= 7.909196 AND  
mean_accel_y <= 2.113771 AND  
mean_accel_z <= 6.317671 AND  
max_accel_z > 0.462738 AND
```

```
min_gyro_smv > 0.062699 AND  
mean_accel_y <= -6.17728 AND  
sma_magn <= 85.987549 AND  
mean_magn_smv <= 47.205959: a (1651.0/2.0)
```

- Javascript

```
if (std_accel_smv > 6.606645 &&  
    std_accel_z <= 5.947899 &&  
    sma_gyro > 0.562692 &&  
    std_accel_y <= 7.909196 &&  
    mean_accel_y <= 2.113771 &&  
    mean_accel_z <= 6.317671 &&  
    max_accel_z > 0.462738 &&  
    min_gyro_smv > 0.062699 &&  
    mean_accel_y <= -6.17728 &&  
    sma_magn <= 85.987549 &&  
    mean_magn_smv <= 47.205959) {  
    return "a"  
}
```

5.7 Σύγκριση με άλλες ερευνητικές εργασίες

Σε αυτό το σημείο, θα συγκριθεί η παρούσα πτυχιακή εργασία με αντίστοιχες ερευνητικές. Αξίζει βέβαια να σημειωθεί ότι η διαδικασία αυτή, δηλαδή η σύγκριση της έρευνας με άλλες αντίστοιχες που ανήκουν στον ίδιο ερευνητικό χώρο, είναι ιδιαίτερα σημαντική, μιας και μπορούν να διακριθούν οι διαφορές καθώς και να επιβεβαιωθούν αποτελέσματα και οι προσεγγίσεις και για τις δυο πλευρές.

Έτσι λοιπόν, η πρώτη ερευνητική εργασία με την οποία θα συγκρίνουμε την παρούσα είναι η έρευνα με όνομα “*Human Activity Recognition from the Acceleration Data of a Wearable Device. Which Features Are More Relevant by Activities?*”[28]. Σε αυτή την έρευνα αντικείμενο μελέτης των ερευνητών ήταν η κατηγοριοποίηση των καθημερινών καταστάσεων και κινήσεων του ανθρώπινου σώματος, χρησιμοποιώντας φορητές συσκευές για την καταγραφή των κινήσεων.

Στην έρευνα αυτή, συμμετείχαν 141 προπτυχιακοί φοιτητές και οι καταγραφές πραγματοποιούνταν σε ελεγχόμενο περιβάλλον, ακολουθώντας συγκεκριμένες οδηγίες. Τα δεδομένα που συλλέχθηκαν ήταν μόνο από τον αισθητήρα του επιταχυνσιόμετρου. Συνολικά εξετάστηκαν 18 διαφορετικές δραστηριότητες οι οποίες ήταν χωρισμένες σε 6 διαφορετικά σενάρια. Το πρώτο σενάριο ήταν η προσωπική περιποίηση, η οποία περιλάμβανε τις δραστηριότητες του χτενίσματος, του πλυσίματος των χεριών και του βουρτσίσματος των δοντιών. Το δεύτερο σενάριο, περιλάμβανε την αερόβια γυμναστική και περιείχε τις δραστηριότητες περπάτημα, τρέξιμο και σκαλιά. Η τρίτη κατηγορία αφορούσε τις δραστηριότητες που περιλαμβάνονται κατά το καθάρισμα του σπιτιού. Περιείχε τις δραστηριότητες

του σιδερώματος των ρούχων, το καθάρισμα των παραθύρων, καθώς και το πλύσιμο των πιάτων. Συνεχίζοντας στην τέταρτη κατηγορία, οι δραστηριότητες εδώ αφορούσαν τη γυμναστική με βάρη και περιλάμβανε τις ασκήσεις διάτασης των χεριών, άρσεις θανάτου και πλευρικές ανατάσεις των χεριών. Η πέμπτη δραστηριότητα αφορούσε τη γυμναστική με τη γενική της έννοια και περιείχε τις δραστηριότητες της πάσας της μπάλας, της αναπήδησης και της αποδοχής της μπάλας. Έκτη και τελευταία κατηγορία ήταν η προετοιμασία φαγητού. Η κατηγορία αυτή περιλάμβανε το ανακάτεμα του φαγητού εντός ενός μπολ, το κόψιμο λαχανικών και το κοσκίνισμα αλευριού.

Παρακάτω ακολουθεί ο πίνακας που περιγράφει τις κατηγορίες και τις δραστηριότητες τους, καθώς και τον αριθμό των συλλεγμένων αρχείων για την κάθε μια από τις 18 διαφορετικές δραστηριότητες.

Scenarios	Activity	Activity id	No. of Instances
Self-Care	Hair grooming	1	577
	Washing hands	2	551
	Teeth brushing	3	527
Exercise (Cardio)	Walking	4	491
	Jogging	5	510
	Stepping	6	500
House cleaning	Ironing	7	579
	Window washing	8	555
	Dish washing	9	577
Exercise (Weights)	Arm curls	10	516
	Deadlift	11	469
	Lateral arm raises	12	511
Sport	Pass	13	627
	Bounce	14	563
	Catch	15	598
Food Preparation	Mixing food in a bowl	16	498
	Chopping vegetables	17	475
	Sieving flour	18	488
Total			9612

Πίνακας 11. Οι κατηγορίες της πρώτης συγκρινόμενης έρευνας [28]

Συνεχίζοντας, τα δεδομένα από τον αισθητήρα του επιταχυνσιόμετρου συλλέγονταν 51.2 φορές το δευτερόλεπτο, δηλαδή, με συχνότητα 51.2Hz για κάθε έναν από τους τρεις άξονες, x, y και z. Για κάθε δραστηριότητα, χρησιμοποιήθηκε ένα παράθυρο χρόνου της τάξης των τεσσάρων δευτερολέπτων.

Παρακάτω ακολουθεί ο πίνακας με τα χαρακτηριστικά τα οποία υπολογίστηκαν.

Feature No.	Feature Name	Feature Description
1-4	Mean value	Mean value of the x, y, z and SMV in the window.
5-8	Maximum	Maximum value of the x, y, z and SMV in the window.
9-12	Minimum	Minimum value of the x, y, z and SMV in the window.
13-16	Standard Deviation	Standard deviation of the samples x, y, z and SMV in the window.
17-20	Range	Range of the samples of SMV in the window.
21-24	Root Mean Square	Root Mean Square of the values of x, y, z and SMV in the window.
25	Signal Magnitude area	Signal Magnitude Area (SMA) across the acceleration signal in x, y and z axis.
26	Spectral Entropy	The normalized information entropy magnitudes of the discrete FFT components of the signal.
27	Total Energy	Sum of the squared magnitudes of the discrete FFT components of the signal

Πίνακας 12. Οι μετρικές που χρησιμοποιήθηκαν από την πρώτη συγκρινόμενη έρευνα [28]

Τέλος, όσον αφορά τους αλγορίθμους που εξετάστηκαν προκειμένου να επιτευχθεί η αναγνώριση των δραστηριοτήτων, αυτοί που δοκιμάστηκαν ήταν οι, Decision Table (DT), Naive Bayes, KNN, Multy Layer Perceptron και Support Vector Machines. Τα αποτελέσματα της έρευνας έδειξαν ότι ο αλγόριθμος με τα καλύτερα αποτελέσματά ήταν ο KNN για όλες τις κατηγορίες δραστηριοτήτων.

Έχοντας αναλύσει λοιπόν την παραπάνω έρευνα, σε αυτό το σημείο, θα τη συγκρίνουμε με την παρούσα πτυχιακή εργασία.

Αρχικά, στην παρούσα εργασία η συμμετοχή των ατόμων σε αριθμό ήταν 8, εκ των οποίων οι 5 ήταν προπτυχιακοί φοιτητές. Στην περίπτωση της καταγραφής και του περιβάλλοντος εντός της, δεν υπήρχε κάποιος περιορισμός ή κάποια συγκεκριμένη οδηγία προκειμένου οι χρήστες να εφαρμόσουν τις δραστηριότητες.

Οι δραστηριότητες ήταν το περπάτημα, το τρέξιμο, η ποδηλασία, η μετακίνηση με το αυτοκίνητο, καθώς και οι μετακινήσεις με το αστικό λεωφορείο και το τρένο. Και στην παρούσα έρευνα χρησιμοποιήθηκαν χρονικά παράθυρα της τάξης των τεσσάρων δευτερολέπτων, ωστόσο, χρησιμοποιήθηκαν τρεις διαφορετικοί αισθητήρες όπου και για τους τρεις, καταγράφηκαν οι άξονες x, y και z. Τα χαρακτηριστικά τα οποία εξήχθησαν ήταν σχεδόν τα ίδια με τη συγκρινόμενη έρευνα με τη διαφορά ότι όλα υπολογίστηκαν για κάθε άξονα κάθε αισθητήρα. Τα δυο χαρακτηριστικά που δε χρησιμοποιήθηκαν ήταν τα τελευταία δυο, δηλαδή, η φασματική εντροπία (Spectral Entropy) και η ολική ενέργεια (Total Energy).

Έχοντας λάβει υπόψη αυτά λοιπόν, καταλήγουμε και στους αλγορίθμους. Στη δική μας περίπτωση, δοκιμάστηκαν οι Naive Bayes, KNN, J48 και PART. Ο αλγόριθμος με τα καλύτερα αποτελέσματα ήταν ο KNN αλλά λόγω της φύσης της

έρευνας, δηλαδή το γεγονός πως έπρεπε το μοντέλο να τρέχει σε πραγματικό χρόνο σε κινητές συσκευές, επιλέχθηκε ο αλγόριθμος PART.

Η δεύτερη σύγκριση που θα γίνει, είναι με την έρευνα “*A Public Domain Dataset for Human Activity Recognition Using Smartphone*”[29]. Σε αυτή την έρευνα, μια ομάδα 30 εθελοντών από ηλικίες 19 έως και 48 επιλέχθηκαν για να ακολουθήσουν ένα πρωτόκολλο από δραστηριότητες, χρησιμοποιώντας ένα έξυπνο κινητό τηλέφωνο το οποίο ήταν δεμένο στη μέση τους. Οι δραστηριότητες που πραγματοποιήθηκαν έγιναν σε ελεγχόμενο περιβάλλον και ήταν συνολικά 6.

Η πρώτη δραστηριότητα ήταν η ακινησία σε όρθια στάση, η δεύτερη καθήμενος, η τρίτη ξαπλωμένος, η τέταρτη περπατώντας και η πέμπτη με την έκπη, ήταν η ανάβαση και η κατάβαση σκαλιών αντίστοιχα. Κάθε εθελοντής πραγματοποίησε τις δραστηριότητες αυτές δυο φορές. Την πρώτη φορά το κινητό τηλέφωνο ήταν δεμένο στη μέση των εθελοντών ενώ τη δεύτερη, το κρατούσαν όπως οι ίδιοι επιθυμούσαν. Κατά τη διαδικασία μετάβασης από τη μια δραστηριότητα στην άλλη, οι εθελοντές ξεκουράζονταν για 5 δευτερόλεπτα.

Παρακάτω ακολουθούν οι δραστηριότητες που εκτελέστηκαν καθώς και ο χρόνος για τον οποίο εκτελούνταν η κάθε μια από αυτές.

No.	Static	Time (sec)	No.	Dynamic	Time (sec)
0	Start (Standing Pos)	0	7	Walk (1)	15
1	Stand (1)	15	8	Walk (2)	15
2	Sit (1)	15	9	Walk Downstairs (1)	12
3	Stand (2)	15	10	Walk Upstairs (2)	12
4	Lay Down (1)	15	11	Walk Downstairs (1)	12
5	Sit (2)	15	12	Walk Upstairs (2)	12
6	Lay Down (2)	15	13	Walk Downstairs (3)	12
			14	Walk Upstairs (3)	12
			15	Stop	0
				Total	192

Πίνακας 13. Οι δραστηριότητες που πραγματοποιήθηκαν από τη δεύτερη συγκρινόμενη έρευνα [29]

Οι αισθητήρες που χρησιμοποιήθηκαν στην έρευνα ήταν το επιταχυνσιόμετρο και το γυροσκόπιο. Και για τους δυο αισθητήρες τα σήματα ήταν τριαξονικά, δηλαδή, και για τον άξονά x, τον y και τον z. Η συχνότητα συλλογής των δεδομένων ήταν τα 50Hz, δηλαδή, αποθηκευόταν οι μεσήςεις των αισθητήρων 50 φορές ανά δευτερόλεπτο. Τα σήματα, υπέστησαν επεξεργασία για τη μείωση θορύβου με ένα διάμεσο φίλτρο και με ένα φίλτρο 3ης σειράς, τύπου χαμηλής διέλευσης. Τα φίλτρα αυτά περιορίσαν τη συχνότητα των σημάτων στα 30Hz. Ωστόσο, αξίζει να σημειωθεί ότι επειδή η κίνησή του ανθρώπινου σώματος είναι κατά 99% κάτω των

15Hz, δεν υπήρξε κάποια επίπτωση στην ακρίβεια των δεδομένων. Το σήμα από τον αισθητήρα του επιταχυνσιόμετρου, το οποίο περιέχει στοιχεία βαρύτητας και κίνησης του σώματος, διαχωρίστηκε χρησιμοποιώντας ένα διαφορετικό φίλτρο χαμηλής διέλευσης, σε επιτάχυνση σώματος και στη βαρύτητα. Η βαρυτική δύναμη θεωρείται ότι έχει μόνο στοιχεία χαμηλής συχνότητας, επομένως βρέθηκε από τα πειράματα ότι τα 0.3Hz ήταν η βέλτιστη συχνότητα για ένα σήμα βαρύτητας.

Τα σήματα χρόνου στη συνέχεια δειγματοίστηκαν σε συρόμενα παράθυρα σταθερού πλάτους 2,56 δευτερολέπτων χρησιμοποιώντας 50% αλληλεπικάλυψη μεταξύ τους. Αυτό, καθώς το περπάτημα ενός μέσου ανθρώπου κυμαίνεται από τα 90 έως τα 130 βήματα το λεπτό, τουλάχιστον ένας κύκλος βημάτων (δυο βήματα) έπρεπε να παρίσταται για κάθε χρονικό παράθυρο έτσι ώστε, άνθρωποι με κινητικές δυσκολίες, όπως οι ηλικιωμένοι, να μπορούν να επωφεληθούν από τη μελέτη αυτή. Τέλος, τα σήματα μετατράπηκαν και στον τομέα της συχνότητάς μέσω μιας συνάρτησης Fourier.

Συνεπώς, 17 χαρακτηριστικά αποκτήθηκαν από την παραπάνω ανάλυση τα οποία μπορούν να διακριθούν και στον παρακάτω πίνακα.

Function	Description
mean	Mean value
std	Standard deviation
mad	Median absolute value
max	Largest values in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of the squares
iqr	Interquartile range
entropy	Signal Entropy
arCoeff	Autorregresion coefficients
correlation	Correlation coefficient
maxFreqInd	Largest frequency component
meanFreq	Frequency signal weighted average
skewness	Frequency signal Skewness
kurtosis	Frequency signal Kurtosis
energyBand	Energy of a frequency interval
angle	Angle between two vectors

Πίνακας 14. Τα χαρακτηριστικά που αποκτήθηκαν από τη δεύτερη συγκρινόμενη έρευνα [29]

Πρώτο χαρακτηριστικό ήταν η μέση τιμή, δεύτερο η τυπική απόκλιση, τρίτο και τέταρτο ήταν η μέγιστη και η ελάχιστη τιμή αντίστοιχα. Έκτο ήταν το εμβαδό του

μεγέθους του σήματος, έβδομο η ενέργεια, όγδοο το διατεταρτημοριακό εύρος, ένατο η εντροπία, δέκατο ήταν οι συντελεστές αυτόματης παλινδρόμησης, ενδέκατο ο συντελεστής συσχέτισης, δωδέκατο η μεγαλύτερη συχνότητα του εκάστοτε χρονικού παραθύρου, δέκατο τρίτο η μέση συχνότητα του εκάστοτε χρονικού παραθύρου, δέκατο τέταρτο η ασυμμετρία, δέκατο πέμπτο η κύρτωση, δέκατο έκτο η ενέργεια της συχνότητας και δέκατο έβδομο και τελευταίο, η γωνία μεταξύ δύο σημάτων στον χώρο.

Συνολικά 561 χαρακτηριστικά αποκτήθηκαν για την περιγραφή κάθε χρονικού παραθύρου. Επιπλέον, για την καλύτερη δυνατή αξιολόγηση της απόδοσης του dataset, αυτό, χωρίστηκε σε δύο τυχαία τμήματα όπου το 70% των δεδομένων χρησιμοποιήθηκε για την εκπαίδευση και το υπόλοιπο 30% για τον έλεγχο των αποτελεσμάτων.

Τα αποτελέσματα εξήχθησαν χρησιμοποιώντας τον γνωστό αλγόριθμο στον τομέα της ΑΑΔ, SVM (Support Vector Machines). Συνολικά υπήρξε 96% ακρίβεια. Παρακάτω παραθέεται ο πίνακας με τον πίνακα σύγχυσης.

	WK	WU	WD	ST	SD	LD	Recall
Walking	492	1	3	0	0	0	99%
W. Upstairs	18	451	2	0	0	0	96%
W. Downstairs	4	6	410	0	0	0	98%
Sitting	0	2	0	432	57	0	88%
Standing	0	0	0	14	518	0	97%
Laying Down	0	0	0	0	0	537	100%
Precision	96%	98%	99%	97%	90%	100%	96%

Πίνακας 15. Τα αποτελέσματα που εξήχθησαν από τη δεύτερη συγκρινόμενη έρευνα [29]

Συγκρίνοντας την εργασία αυτή με την παρούσα πτυχιική, διακρίνει κανείς πως η εν λόγω εργασία, χρησιμοποίησε πάρα πολλά χαρακτηριστικά για κάθε χρονικό παράθυρο, σε σχέση με την παρούσα πτυχιική. Επίσης, τα αποτελέσματα που πέτυχαν ήταν εξαιρετικά και αποδεικνύουν πως αν υπάρχουν πάρα πολλά χαρακτηριστικά, τότε η ακρίβεια της εκτίμησης μπορεί να αυξηθεί. Ωστόσο, κάλο θα ήταν να μην ξεχνάει κανείς ότι οι εθελοντές που πραγματοποίησαν αυτές τις δραστηριότητες, ήταν εντός ελεγχόμενου περιβάλλοντος και όχι ελεύθεροι να πραγματοποιήσουν τις δραστηριότητες όπως αυτοί ήθελαν, κάτι που στην παρούσα πτυχιική συνέβη εκ προθέσεως. Όπως παρουσιάζεται και παρακάτω, σκοπός της παρούσας εργασίας ήταν να αντιλαμβανόμαστε αυτόματα, συγκεκριμένες

δραστηριότητες χρηστών, χωρίς να υπάρχει κάποιος όρος ή περιορισμός για την εκτίμηση αυτή.

Η τρίτη και τελευταία σύγκριση που θα γίνει, είναι με την έρευνα *“Transportation Mode Detection from Low-Power Smartphone Sensors Using Tree-Based Ensembles”*[33]. Σε αυτή την έρευνα, συμμετείχαν 27 εθελοντές. Συνολικά πραγματοποιήθηκαν 550 μετακινήσεις, 374 χρησιμοποιώντας ως μέσο μετακίνησης το αυτοκίνητο και 176 χρησιμοποιώντας μόνο τα μέσα μαζικής μεταφοράς. Οι μετακινήσεις που καταγράφηκαν κατά τη διάρκεια της έρευνας ήταν, η μετακίνηση με το τρένο, το αυτοκίνητο, το τραμ και το λεωφορείο. Οι καταγραφές πραγματοποιήθηκαν χρησιμοποιώντας μια εφαρμογή για έξυπνα κινητά τηλέφωνα και οι αισθητήρες που χρησιμοποιήθηκαν ήταν το επιταχυνσιόμετρο και το γυροσκόπιο. Κατά τη διάρκεια των καταγραφών, η θέση της συσκευής ήταν τυχαία.

Για τον επεξεργασία των δεδομένων, χρησιμοποιήθηκαν κυλιόμενα παράθυρα μεγέθους 8 δευτερολέπτων, καθώς επίσης και ενός δευτερολέπτου αναδίπλωση με το αμέσως προηγούμενο και επόμενο χρονικό παράθυρο. Ως χαρακτηριστικά για την εφαρμογή αλγορίθμων τεχνητής νοημοσύνης, δημιουργήθηκαν η ελάχιστη και η μέγιστη τιμή, καθώς και η τυπική απόκλιση, η μέση τιμή, η κλίση και τέλος, η περιστροφή της εκάστοτε συσκευής. Οι αλγόριθμοι τεχνητής νοημοσύνης που εφαρμόστηκαν, ήταν ο Random Forest (RF) και ο Gradient Boosting Machine (GBM). Στη συνέχεια, τα δεδομένα χωρίστηκαν σε 4 κύριες ομάδες. (1) όλες οι πληροφορίες της διαδρομής, (2) όλες οι πληροφορίες της διαδρομής εκτός του χρόνου που συμπεριλάμβανε τη με χρήση του κινητού, (3) όλες οι πληροφορίες της διαδρομής εκτός του χρόνου που συμπεριλάμβανε στάσεις και (4) όλες οι πληροφορίες της διαδρομής εκτός του χρόνου που συμπεριλάμβανε τη χρήση κινητού και στάσεων.

Variable	Meta-variables	Name
Acceleration	Minimum	accMIN
	Maximum	accMAX
	Standard deviation	accSD
	Median	accMED
	Mean	accMEAN
Pitch	Minimum	pitMIN
	Maximum	pitMAX
	Standard deviation	pitSD
	Median	pitMED
	Mean	pitMEAN
Roll	Minimum	rolMIN
	Maximum	rolMAX
	Standard deviation	rolSD
	Median	rolMED
	Mean	rolMEAN
Angular Velocity	Minimum	gyrMIN
	Maximum	gyrMAX
	Standard deviation	gyrSD
	Median	gyrMED
	Mean	gyrMEAN
Speed	Smoothened speed based on moving average filter	speed
Driver	A unique number assigned to every user in the database	driver

Πίνακας 16. Τα χαρακτηριστικά που εξήχθησαν από την τρίτη συγκρινόμενη έρευνα [33]

	RF (%)	GBM (%)
Sensitivity	74.4	69.2
Specificity	93.0	94.4
Precision	85.3	87.1
NPV	86.8	84.8
Accuracy	86.4	85.5

Πίνακας 17. Τα αποτελέσματα των αλγορίθμων από την τρίτη συγκρινόμενη έρευνα για τα δεδομένα “όλες οι πληροφορίες του ταξιδιού” [33]

	Sensitivity (%)	Specificity (%)	Precision (%)	NPV (%)	Accuracy (%)
RF					
Without MU	71.80	93.00	84.80	85.70	85.50
Without STOPS	69.20	94.40	87.10	84.80	85.50
Without MU + without STOPS	78.80	92.00	81.30	90.80	88.00
GBM					
Without MU	64.10	94.40	86.20	82.70	83.60
Without STOPS	71.80	95.80	90.30	86.10	87.30
Without MU + without STOPS	78.80	93.30	83.90	90.90	88.90

Πίνακας 18. Τα αποτελέσματα των αλγορίθμων από την τρίτη συγκρινόμενη έρευνα για τα δεδομένα “όλες οι πληροφορίες του ταξιδιού εκτός του χρόνου που συμπεριλάμβανε τη χρήση κινητού και στάσεων” [33]

Συγκρίνοντας την έρευνα αυτή με την παρούσα πτυχιική, διακρίνει κανείς πως η εν λόγω έρευνα επικεντρώθηκε κυρίως, στην αναγνώριση του μέσου μετακίνησης καθ' όλη τη διάρκεια της εκάστοτε μετακίνησης και όχι στην εναλλαγή των μετακινήσεων, όπως η εργασία ετούτη. Επίσης, κατά την ανάλυση των δεδομένων, η συγκρινόμενη εργασία έκανε παραδοχές, δηλαδή χρησιμοποιήθηκαν δεδομένα όλης της διαδρομής και δεδομένα όλης της διαδρομής αλλά, χωρίς όμως τις στάσεις και τον χρόνο χρήσης του κινητού κατά τη μετακίνηση αυτή. Κατά αυτόν τον τρόπο, η αναγνώριση της μέσης δραστηριότητας ενός ατόμου είναι αδύνατον να υπολογιστεί κατά τη διάρκεια της ημέρας ενώ αυτός πραγματοποιεί μετακινήσεις τόσο με τα πόδια όσο και με όλα τα υπόλοιπα πιθανά μέσα μεταφοράς, όπως αυτά έχουν αναλυθεί παραπάνω, κατά την ανάλυσή. Τα αποτελέσματα βέβαια από τα μοντέλα, ήταν πολύ κοντά στα αποτελέσματα τούτης της πτυχιικής, πράγμα που

επιβεβαιώνει την εγκυρότητα και την ακρίβεια τους. Τέλος, η ερευνητική ομάδα, δεν ανέπτυξε κάποια εφαρμογή για την πρακτική εφαρμογή των εξαγόμενων μοντέλων, προκείμενου να επιβεβαιωθούν τα αποτελέσματα της έρευνας.

6. Υλοποίηση και Οθόνες

Σε αυτή την ενότητα θα αναλυθούν τα πιθανά σενάρια χρήσης των δυο εφαρμογών μαζί με τις αντίστοιχες οθόνες τους, καθώς και θα παρουσιαστούν όλες οι ενδεικτικές οθόνες για την κάθε εφαρμογή.

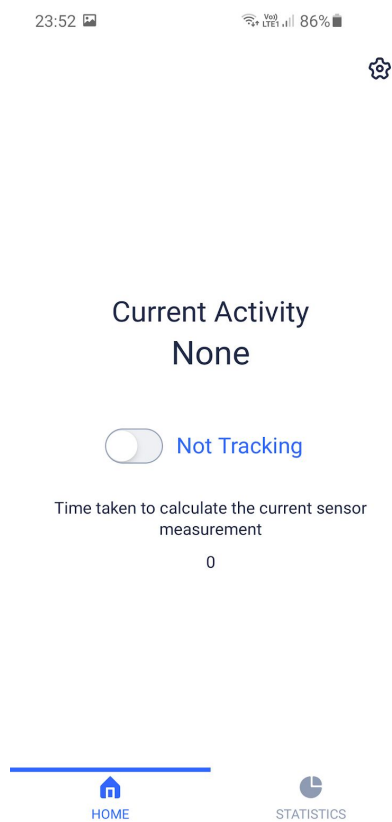
6.1 Σενάριο χρήσης και ενδεικτικές οθόνες για την εφαρμογή αυτόματης εκτίμησης του μέσου μετακίνησης

Σε αυτό το σημείο θα εξεταστεί ένα σενάριο χρήσης της εφαρμογής. Ουσιαστικά, θα είναι σαν μια καταγραφή της οθόνης του κινητού, απλά υπό τη μορφή κειμένου και εικόνων.

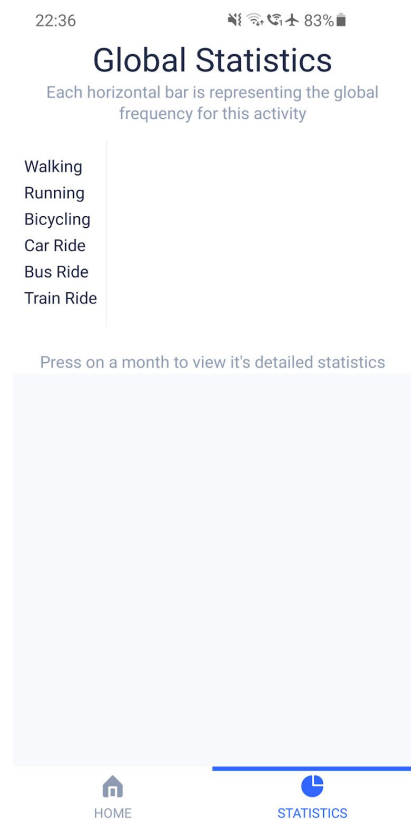
Έτσι λοιπόν, ένα υποθετικό σενάριο είναι το εξής. Ο Νίκος, φοιτητής του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου, φεύγει από το σπίτι του στη Γαρίτσα και αποφασίζει να χρησιμοποιήσει για πρώτη φορά μια εφαρμογή που του πρότειναν στη σχολή, έτσι, ανοίγει την εφαρμογή (βλ. Οθόνη 1) και μεταβαίνει στα στατιστικά στοιχεία (οθόνη 2). Όμως επειδή είναι η πρώτη φορά που χρησιμοποιεί την εφαρμογή, δεν υπάρχουν δεδομένα προς εμφάνιση. Οπότε κατευθύνεται πάλι στην αρχική οθόνη (βλ. Οθόνη 3) και ξεκινάει τη διαδικασία καταγραφής, ωστόσο, δεν του αρέσει το φωτεινό θέμα εμφάνισης που έχει η εφαρμογή, οπότε κατευθύνεται στις ρυθμίσεις (βλ. Οθόνη 4 & 5) και το αλλάζει σε σκούρο, έπειτα, επιστρέφει στην αρχική οθόνη της εφαρμογής, εξέρχεται αυτής και κλείνει την οθόνη του κινητού του. Όταν φτάνει στη σχολή, ανοίγει την εφαρμογή (βλ. Οθόνη 6) και σταματά τη διαδικασία καταγραφής (βλ. Οθόνη 7). Έτσι λοιπόν, παρατηρεί τα δεδομένα που κατέγραψε η εφαρμογή (βλ. Οθόνη 8) και επιλέγει τον μήνα που περιέχει την καταγραφή που μόλις πραγματοποίησε. Επιλέγει την ημέρα της καταγραφής (βλ. Οθόνη 9) και βλέπει όλες τις εκτιμήσεις για κάθε χρονικό παράθυρο 4 δευτερολέπτων (βλ. Οθόνη 10). Έπειτα αποφασίζει ότι η συγκεκριμένη καταγραφή δε θα έπρεπε να υπάρχει και σβήνει όλες τις εκτιμήσεις (βλ. Οθόνη 11). Ύστερα, επιστρέφει στην προηγούμενη οθόνη (βλ. Οθόνη 12) μετά, στην προηγούμενη αυτής (βλ. Οθόνη 13) και τέλος στην αρχική οθόνη της εφαρμογής (βλ. Οθόνη 14).

Επίσης, αξίζει να σημειωθεί το εξής. Η εναλλαγή του μέσου μετακίνησης γίνεται με τρόπο αυτόματο χωρίς να απαιτείται διάδραση με τον χρήστη. Εξ' άλλου,

αυτός ήταν και αρχικός και κύριος σκοπός της εφαρμογής, να καταγράφει όλες τις μετακινήσεις ενός χρήστη, όσο το δυνατόν απρόσκοπτα γίνεται.

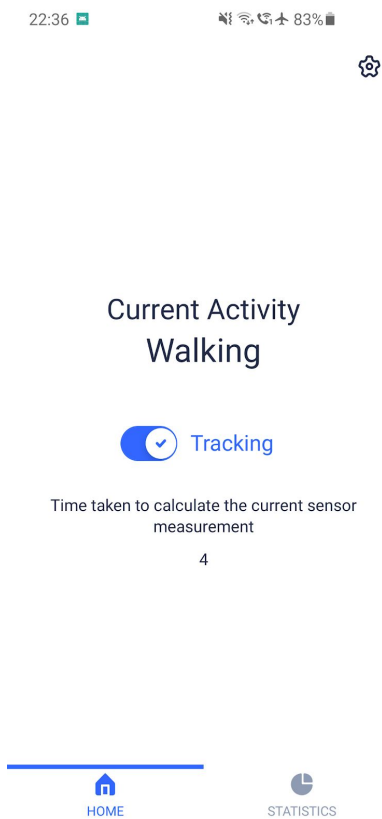


Οθόνη 1

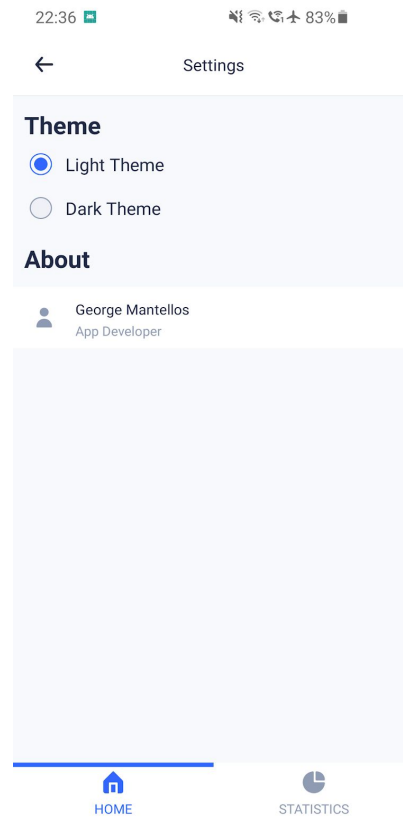


Οθόνη 2

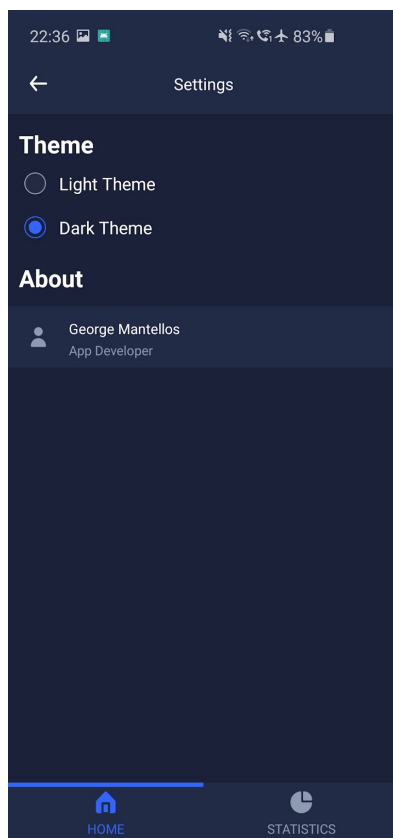




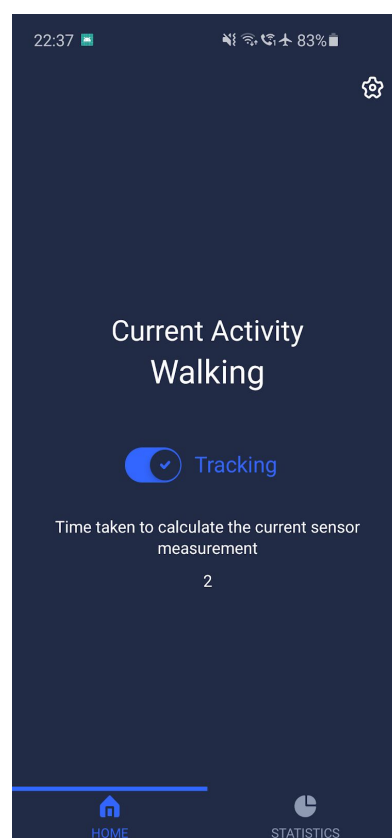
Οθόνη 3



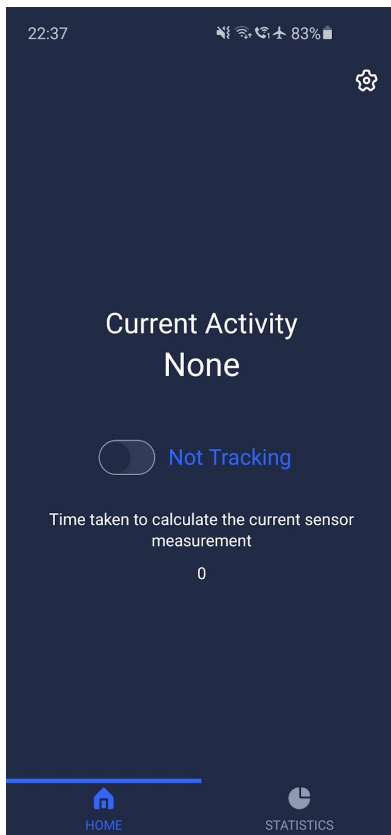
Οθόνη 4



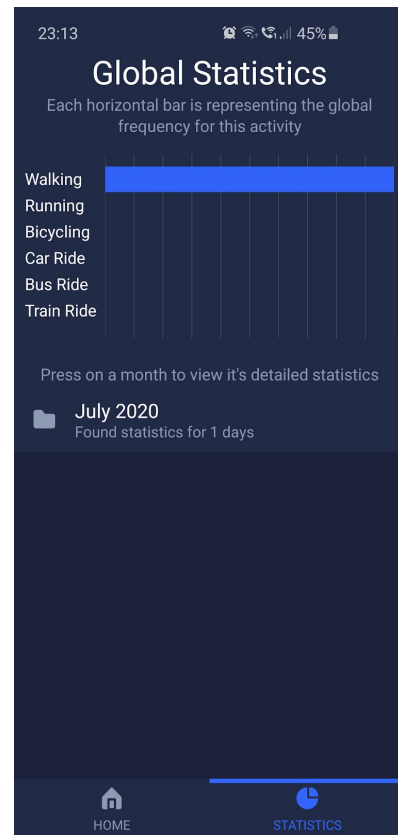
Οθόνη 5



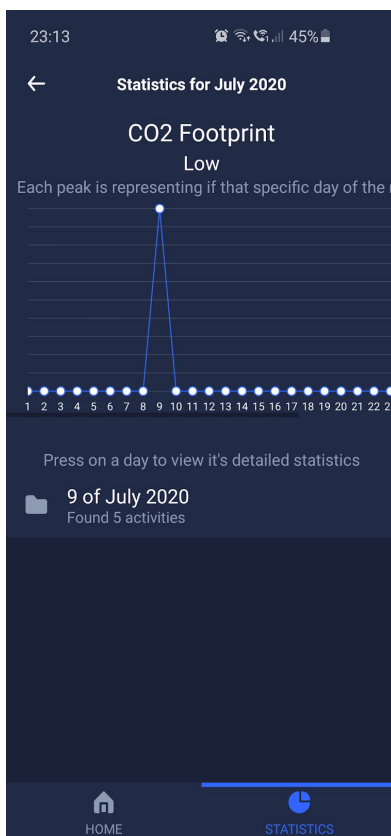
Οθόνη 6



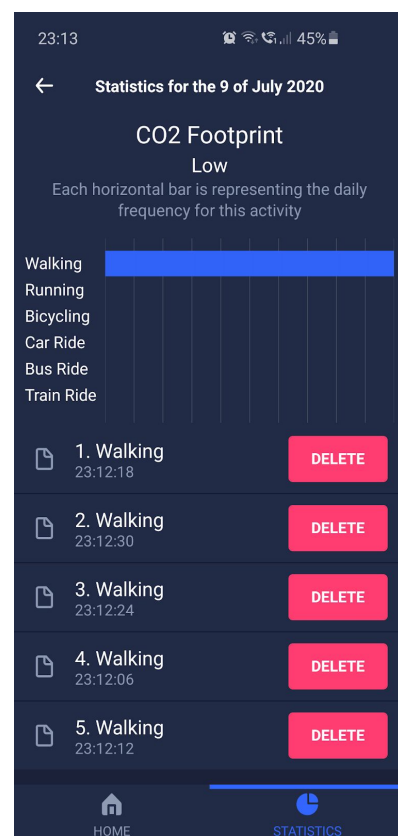
Οθόνη 7



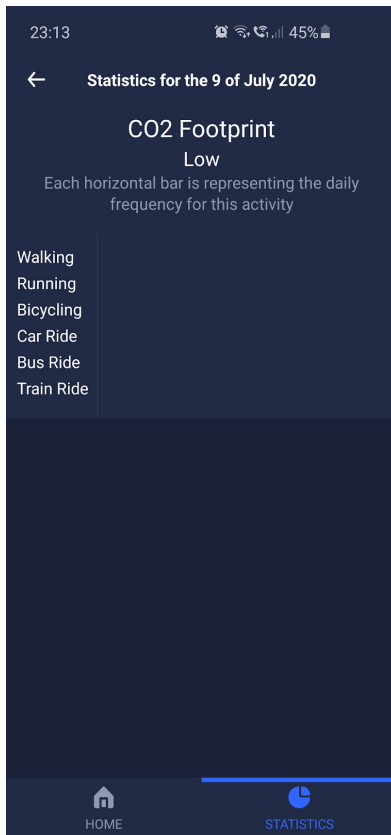
Οθόνη 8



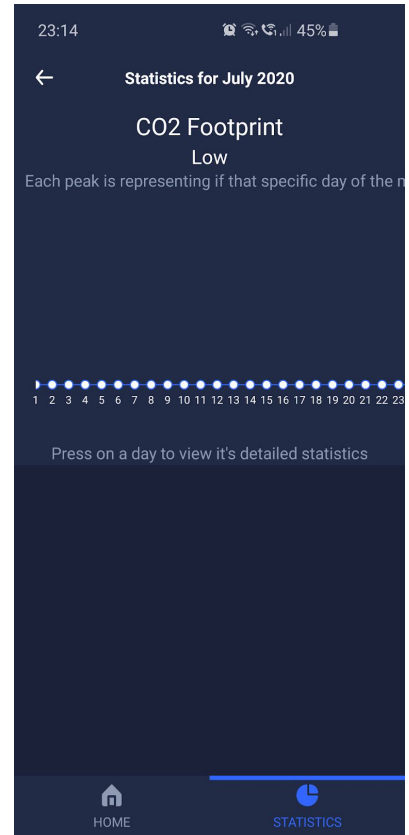
Οθόνη 9



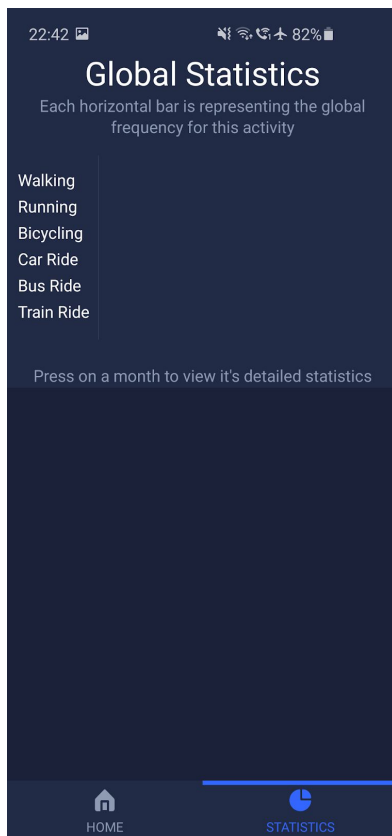
Οθόνη 10



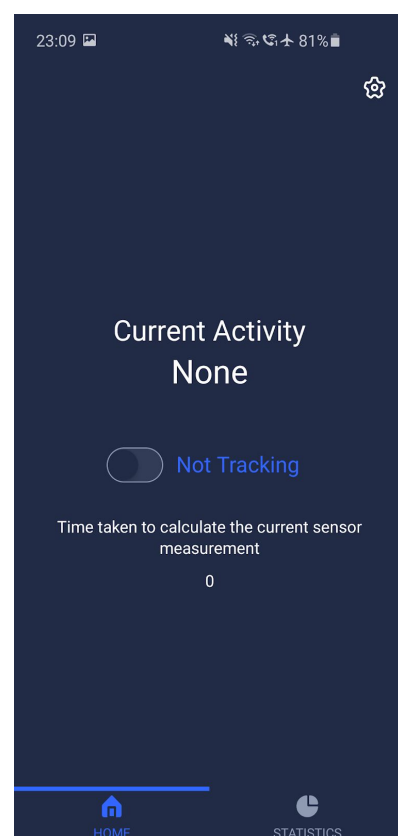
Οθόνη 11



Οθόνη 12



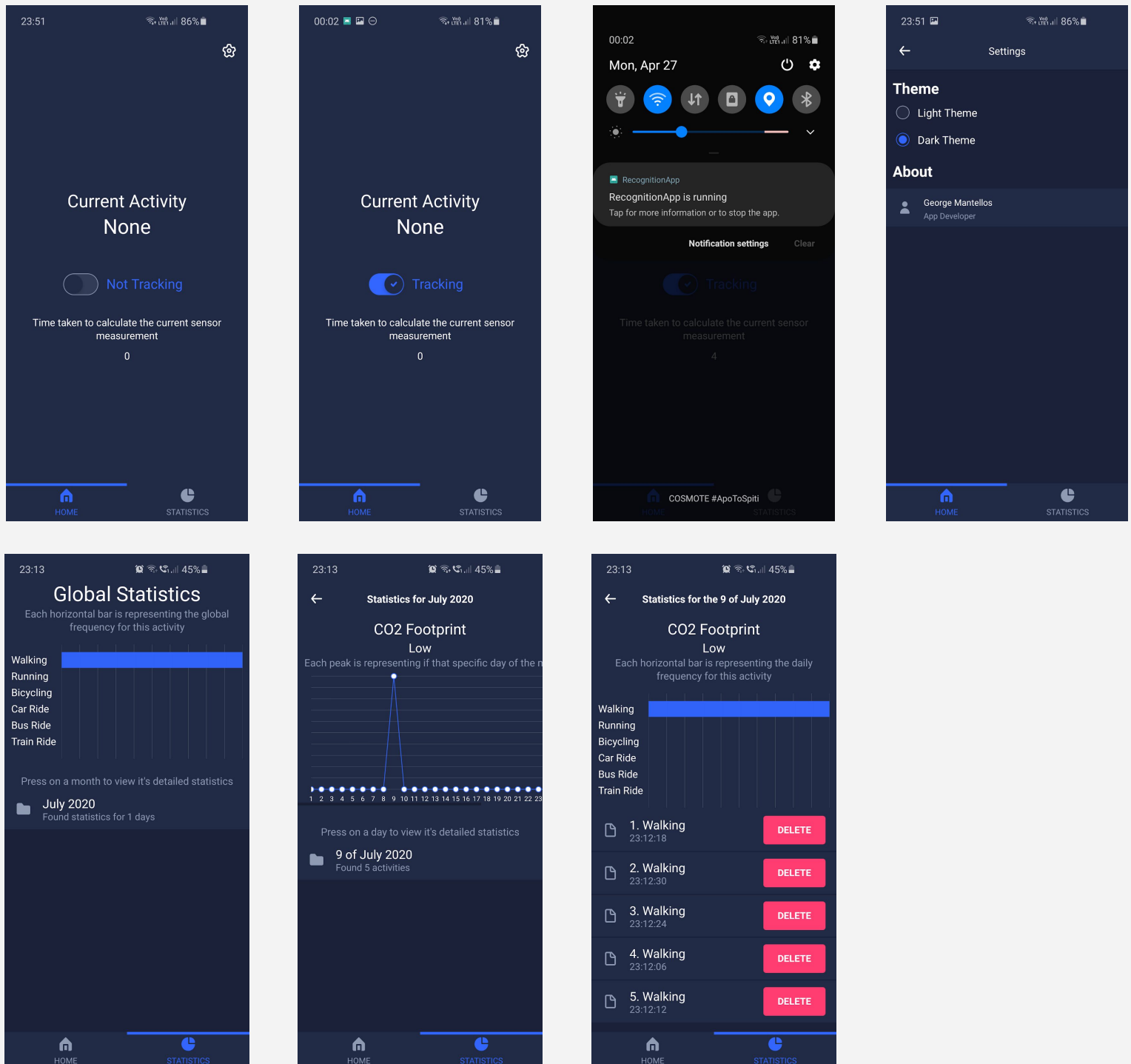
Οθόνη 13



Οθόνη 14

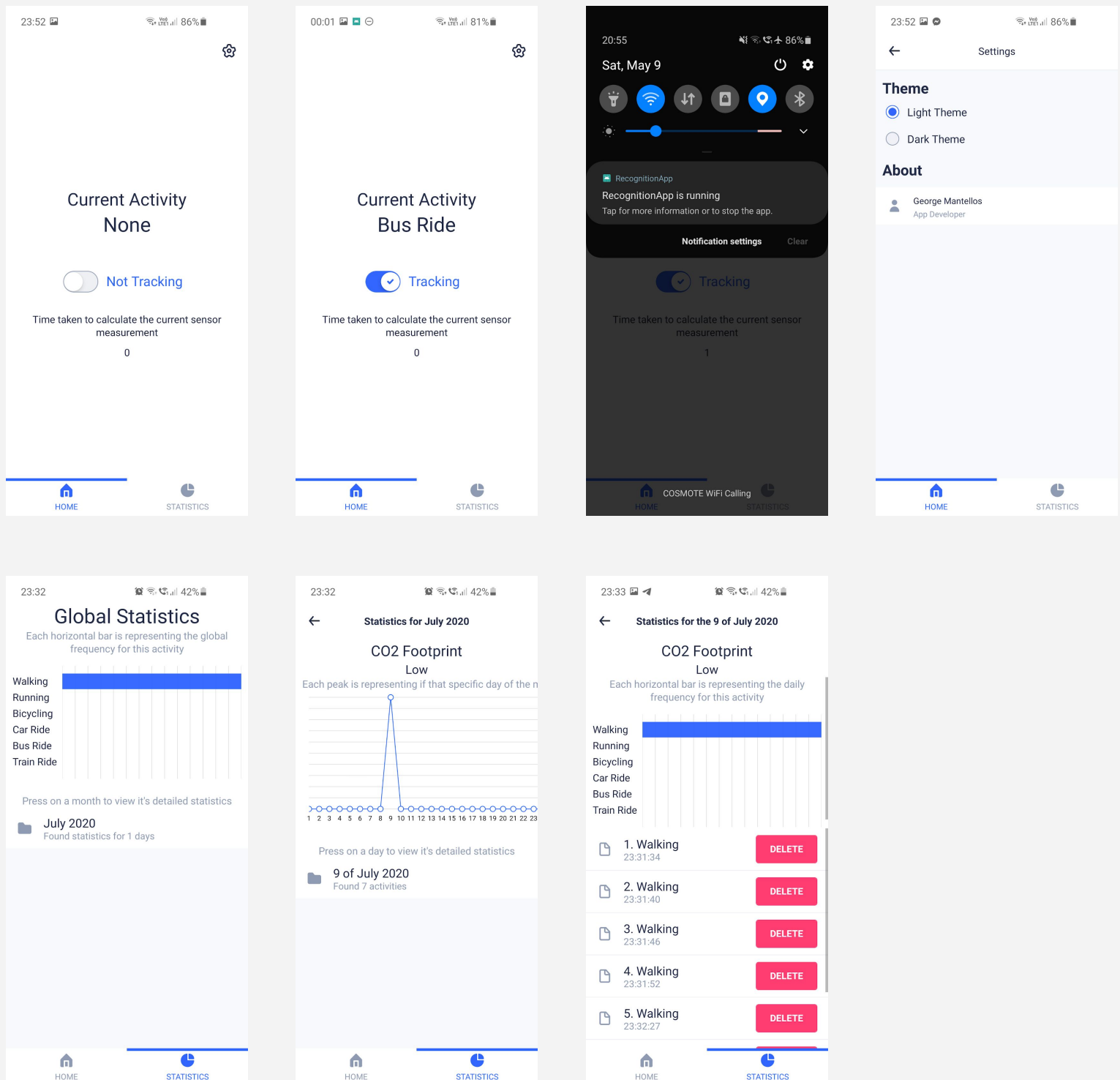
6.1.1 Σκούρο θέμα εμφάνισης

Παρακάτω παραθέτονται όλες οι οθόνες της εφαρμογής έχοντας ως επιλογή για θέμα εμφάνισης το σκούρο.



6.1.2 Φωτεινό θέμα εμφάνισης

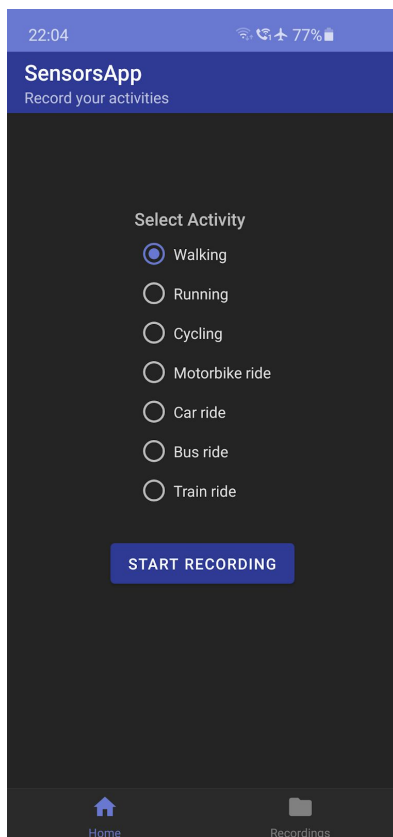
Παρακάτω παραθέτονται όλες οι οθόνες της εφαρμογής έχοντας ως επιλογή για θέμα εμφάνισης το φωτεινό.



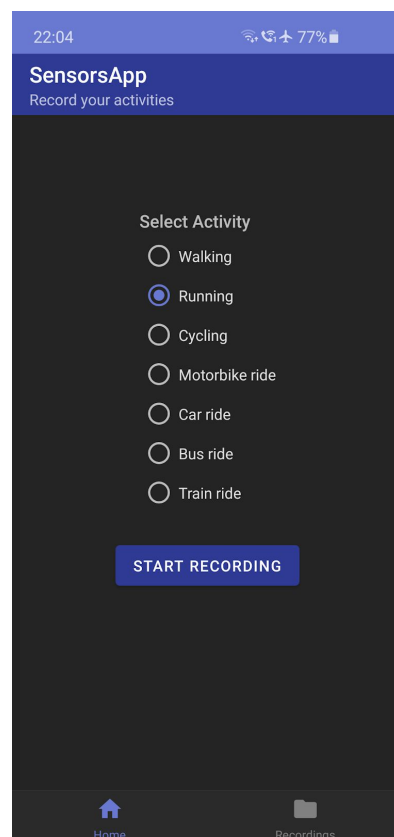
6.2 Σενάριο χρήσης και ενδεικτικές οθόνες για την εφαρμογή συλλογής των δεδομένων

Σε αυτό το σημείο θα εξεταστεί ένα σενάριο χρήσης της εφαρμογής συλλογής δεδομένων SensorsApp. Ουσιαστικά, θα είναι σαν μια καταγραφή της οθόνης του κινητού, όπως ακριβώς παρουσιάστηκε και για την εφαρμογή αυτόματης εκτίμησης, απλά υπό τη μορφή κειμένου και εικόνων.

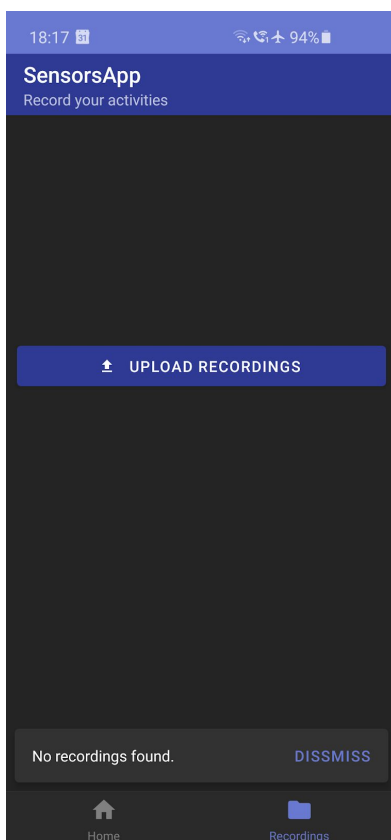
Αρχικά, ο χρήστης ανοίγει την εφαρμογή και βρίσκεται στην κυρία οθόνη της (βλ. Οθόνη 1). Έπειτα αλλάζει την προεπιλεγμένη μετακίνηση (μετακίνηση με τα πόδια) και επιλέγει τη μετακίνηση που είναι για τρέξιμο (βλ. Οθόνη 2). Ύστερα, χωρίς να έχει ξεκινήσει την καταγραφή, κατευθύνεται στην οθόνη η οποία είναι υπεύθυνη για το ανέβασμα των καταγραφών στο cloud, ωστόσο, επειδή είναι η πρώτη φορά που χρησιμοποιεί την εφαρμογή, προφανώς, δεν υπάρχουν καταγραφές προς ανέβασμα (βλ. Οθόνη 3). Έτσι, κατευθύνεται πάλι στην προηγούμενη οθόνη και ξεκίνα την καταγραφή για την επιλεγμένη μετακίνηση (βλ. Οθόνη 5). Στη συνέχεια, βλέπει την ειδοποίηση που δημιουργήθηκε αυτόματα, η οποία σηματοδοτεί την κατάσταση της καταγραφής, και κλείνει την οθόνη της συσκευής του προκειμένου να ξεκινήσει η καταγραφή. Σε αυτό το σημείο, ας υποθέσουμε πως πέρασε το χρονικό διάστημα για το οποίο ο χρήστης ήθελε να καταγράψει οπότε ανοίγει την εφαρμογή και πατάει “STOP RECORDING” (βλ. Οθόνη 6). Έπειτα, κατευθύνεται στην οθόνη η οποία είναι υπεύθυνη για το ανέβασμα των καταγραφών στο cloud και ανεβάζει την καταγραφή (βλ. Οθόνη 7).



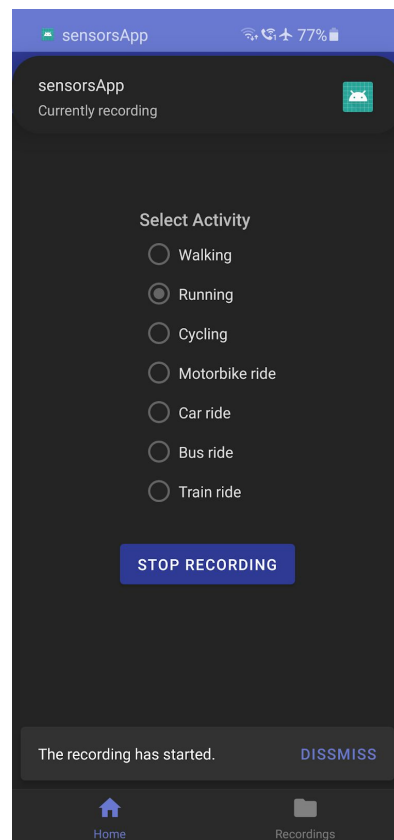
Οθόνη 1



Οθόνη 2

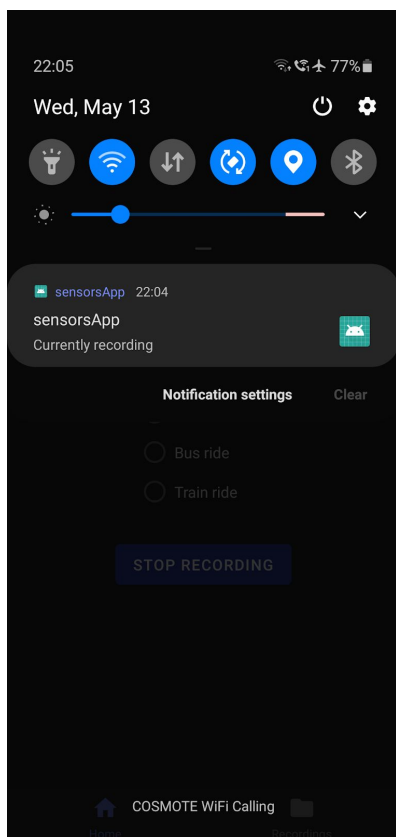


Οθόνη 3

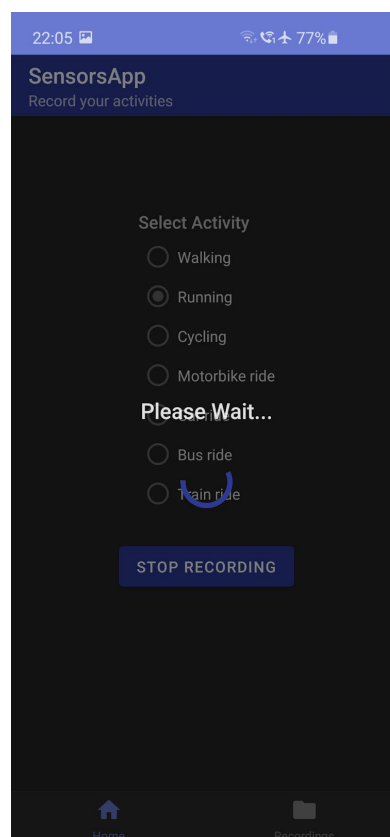


Οθόνη 4

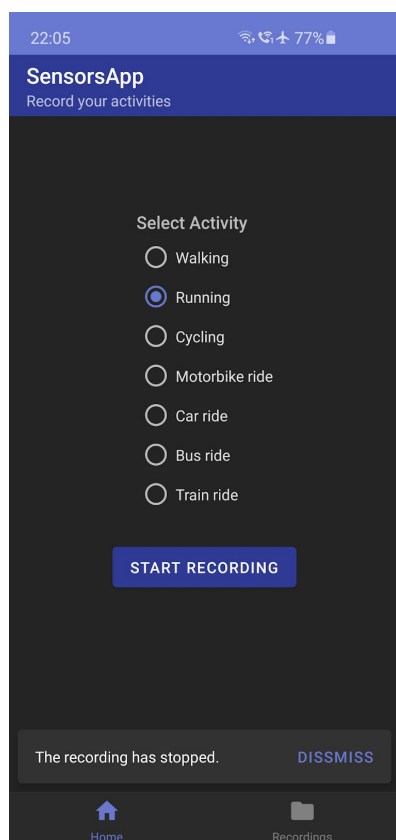




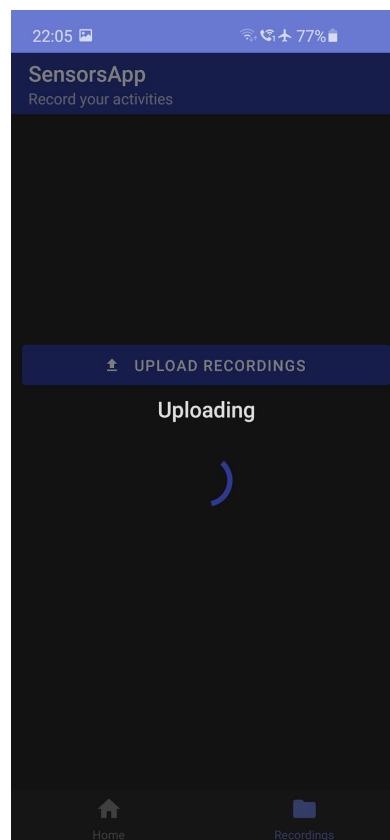
Οθόνη 5



Οθόνη 6



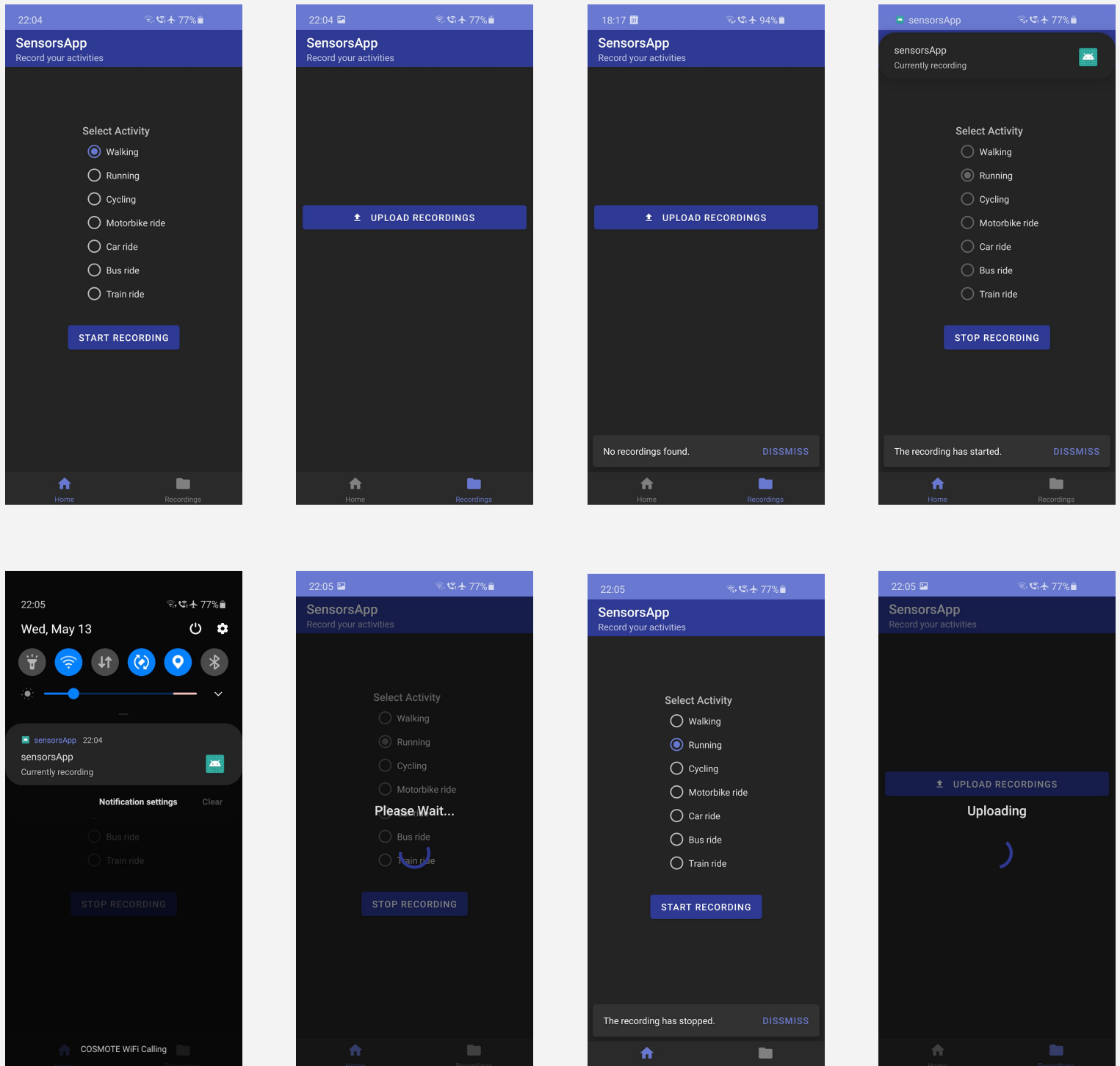
Οθόνη 7



Οθόνη 8

6.2.1 Θέμα εμφάνισης

Παρακάτω παραθέτονται όλες οι οθόνες της εφαρμογής. Υπενθυμίζεται ότι η εφαρμογή δεν παρέχει τη δυνατότητα αλλαγής του θέματος εμφάνισης



7. Συμπεράσματα

Σε αυτή την ενότητα θα αναλυθούν τα συμπεράσματα τα οποία εξήχθησαν από τη μελέτη και την έρευνα για την πραγματοποίηση αυτής της πτυχιακής εργασίας. Πιο συγκεκριμένα, τα συμπεράσματα χωρίστηκαν σε ερευνητικά, τα οποία αφορούσαν το κομμάτι της αυτόματης εκτίμησης του μέσου μετακίνησης, σε προγραμματιστικά, τα οποία αφορούσαν τις γνώσεις που αποκομίσθηκαν, και σε πιθανές μελλοντικές κατευθύνσεις τις οποίες θα μπορούσε να ακολουθήσει ο αναγνώστης αν θα ήθελε να συνεχίσει την έρευνα.

7.1 Ερευνητικά Συμπεράσματα

Η υπάρχουσα έρευνα απέδειξε ότι υπάρχει συγκεκριμένη διάκριση μεταξύ των διαφόρων τρόπων μεταφοράς που μελετήθηκαν. Είτε πρόκειται για περπάτημα, τρέξιμο, ποδηλασία, μεταφορά με το αυτοκίνητο, το τρένο ή με το αστικό λεωφορείο, είναι σαφές ότι μπορεί να προβλεφθεί αυτόματα κάνοντας χρήση μόνο των αισθητήρων κίνησης. Αυτό συμβαίνει, ως επί το πλείστον, επειδή το ανθρώπινο σώμα έχει, ενώ είναι διαφορετικό για κάθε άνθρωπο, το ίδιο εύρος κινήσεων κατά την εκτέλεση τέτοιων μετακινήσεων. Για παράδειγμα, όταν κάποιος βρίσκεται στο αυτοκίνητο ή στο αστικό λεωφορείο. Ωστόσο, διαφαίνεται ότι υπάρχει μεγάλο περιθώριο για πιθανές βελτιώσεις, όπως αναλύεται και παρακάτω.

7.2 Προγραμματιστικά Συμπεράσματα

Φτάνοντας στο τέλος λοιπόν αυτής της έρευνας, αξίζει να σημειωθούν και τα προγραμματιστικά συμπεράσματα τα οποία απορρέουν.

Εν τέλη, η επιλογή για τη δημιουργία των εφαρμογών οι οποίες προοριζόταν για κινητά τηλέφωνα χρησιμοποιώντας την τεχνολογία React Native αποδείχθηκε σωστή. Επιπλέον, αποδείχθηκε ότι το React Native, παρότι σχετικά καινούριο σαν τεχνολογία και όχι ιδιαίτερα αγαπητό από την κοινότητα προγραμματιστών των native εφαρμογών, είναι απόλυτα ικανό να δημιουργήσει λειτουργικές, εύχρηστες, όμορφες και γρήγορες εφαρμογές πολύ γρήγορα και συγκριτικά με τις native εφαρμογές, πολύ ευκολότερα.

Ένα άλλο προγραμματιστικό συμπέρασμα το οποίο απορρέει από την εργασία ετούτη, είναι ότι δεν είναι αναγκαίο να δημιουργείται πάντα, μια καινούρια και άκρως εξειδικευμένη λύση για να επιλύσει ένα πρόβλημα. Μπορεί κάλλιστα να χρησιμοποιηθεί μια υπάρχουσα τεχνολογία/υπηρεσία ή ο συνδυασμός περισσότερων, προκειμένου να επιταχυνθεί η ανάπτυξή του λογισμικού και να επιτευχθεί το βέλτιστο αποτέλεσμα. Φυσικά, το παραπάνω αναφέρεται στην

υπηρεσία διαμοιρασμού αρχείων Dropbox και στο εξαιρετικό API (Application Programming Interface) που παρέχει, το οποίο επέτρεψε τη δυνατότητα αποστολής των καταγραφών από τα κινητά των χρηστών σε ένα κεντρικό αποθετήριο για τη μαζική συλλογή τους.

7.3 Πιθανές κατευθύνσεις για μελλοντική συνέχεια της έρευνας

Οι πιθανές κατευθύνσεις για μελλοντική συνέχεια της έρευνας που διαφαίνονται από την παρούσα εργασία, επικεντρώνονται κυρίως στη βελτίωση της διαδικασίας της αυτόματης εκτίμησης του μέσου μετακίνησης. Ουσιαστικά, ο τρόπος με τον οποίο πραγματοποιείται η αυτόματη εκτίμηση, είναι με την προσπέλαση ενός συνόλου κανόνων, μέχρις ότου να βρεθεί ο κανόνας που ικανοποιείται από τις στατιστικές μετρικές του παραθύρου για το οποίο εξετάζουμε. Αυτό, αλγοριθμικά εξεταζόμενο, προκύπτει ότι είναι εξαιρετικά αργό, αφού ορισμένοι κανόνες περιέχουν σχεδόν τις ίδιες συνθήκες ελέγχου με άλλους κανόνες. Αυτό, οφείλεται στο δεδομένο ότι ένας κανόνας είναι μια διαδρομή ενός δέντρου, από τη ρίζα έως στο βέλτιστο φύλλο του. Συνεπώς, δυο κανόνες μπορεί να έχουν εξαιρετικά όμοια συνθήκη ελέγχου, αφού έχουν παραχθεί από σχεδόν, την ίδια διαδρομή ενός δέντρου. Ωστόσο, αν αντί για κανόνες, σαν δομή για την αυτόματη εκτίμηση του μέσου μετακίνησης, είχαμε ένα δέντρο, τότε η προσπέλαση του θα ήταν σαφώς γρηγορότερη. Πιο συγκεκριμένα, αυτό που συμβαίνει με την υπάρχουσα λύση είναι το εξής, αφότου υπολογιστούν οι μετρικές για το εκάστοτε χρονικό παράθυρο των τεσσάρων δευτερολέπτων, αρχίζει η διαδικασία εύρεσης του κανόνα που θα εκτιμήσει το μέσο μετακίνησης. Κατά αυτό τον τρόπο, θα πρέπει να δοκιμαστούν, στη χειρότερη περίπτωση, όλοι οι κανόνες, μέχρις ότου βρεθεί αυτός ο ποιος ικανοποιείται. Αυτό είναι εξαιρετικά κοστοβόρο, τόσο σε χρόνο όσο και σε απόδοση, αφού, οι κανόνες είναι παραγμένοι από το ίδιο δέντρο, αυτό σημαίνει ότι δύο γειτονικά φύλλα του δέντρου θα έχουν σχεδόν την ίδια συνθήκη ελέγχου, όταν στη χειρότερη περίπτωση, θα πρέπει να εξεταστούν και οι δυο συνθήκες. Ουσιαστικά, είναι σαν να κάνουμε προσπέλαση ξανά και ξανά το ίδιο δέντρο, κάθε φορά που δε φτάνουμε σε ένα φύλλο. Όμως, αν υπήρχε μια δομή δέντρου και όχι κανόνων εξαγμένων από αυτό, η διαδικασία εύρεσης του φύλλου θα ήταν θεωρητικά, τάξης μεγέθους μικρότερη τόσο για τον χρόνο που χρειάζεται προκειμένου να βρεθεί ένα φύλλο, όσο και σε απόδοση.

Συνεχίζοντας αυτό που θα άξιζε επίσης να ερευνηθεί, είναι ο αριθμός των στατιστικών μετρικών που υπολογίζονται ανά χρονικό παράθυρο. Αυτή τη στιγμή, ο αριθμός των μετρικών που υπολογίζονται είναι 38 και περιλαμβάνει και τους τρεις αισθητήρες. Το ενδεχόμενο πρόβλημα που διαφαίνεται, δεν είναι ο αριθμός των αισθητήρων καθ' αυτός, αλλά ο αριθμός των μετρικών που υπολογίζονται για τον κάθε αισθητήρα αντίστοιχα. Στο τωρινό στάδιο της έρευνας, όποια μετρική

υπολογίζεται για τον αισθητήρα του επιταχυνσιόμετρου, υπολογίζεται και για τους άλλους δυο, δηλαδή για τον αισθητήρα του γυροσκοπίου και τον αισθητήρα του μαγνητόμετρου, ενέχοντας έτσι τον κίνδυνο ότι μια μετρική που είναι χρήσιμη για τον ένα αισθητήρα, μπορεί να είναι εντελώς άχρηστη για τον άλλο. Ο αλγόριθμος που χρησιμοποιήθηκε για τη μείωση των μετρικών, ναι μεν μείωσε τον αριθμό τους βασιζόμενος στην εξάρτηση των μετρικών μεταξύ τους, αλλά από την άλλη, βασίστηκε σε αυτό ακριβώς που προσπαθούμε να διορθώσουμε, δηλαδή, στις αρχικές μετρικές. Αν εξετάζαμε τον κάθε αισθητήρα ξεχωριστά, προκειμένου να μετρήσουμε την απόδοση της κάθε μετρικής που μπορεί να εφαρμοστεί σε αυτόν, τότε ίσως να μπορούσε να μειωθεί ο αριθμός των μετρικών και κατά συνέπεια, ο συνολικός αριθμός τους.

Όσον αφορά την κινητή εφαρμογή αποτυπώματος άνθρακα, στόχος είναι να αναπτυχθεί μια εφαρμογή που θα δημιουργεί εξατομικευμένα γραφήματα, θα ταξινομεί τους χρήστες της και θα τους εμπλέκει σε μια διαδικασία gamification σε συνεργασία με τις τοπικές κοινότητες και δήμους. Μπορεί να χρησιμοποιηθεί ως ένα απλό εργαλείο ως προς την ευαισθητοποίηση για το περιβάλλον και ως προς την καθοδήγηση του πληθυσμού ώστε να ακολουθηθεί ένας πιο βιώσιμος τρόπος ζωής παρακολουθώντας και τελικά μειώνοντας το περιβαλλοντικό του αποτύπωμα.

Βιβλιογραφία

- [1] J.R. Kwapisz, G.M. Weiss, and S.A. Moore. Activity recognition using cell phone accelerometers. SIGKDD Explorations Newsletter, 12(2):74–82, 2011.
- [2] T. Brezmes, J.L. Gorricho, and J. Cotrina. Activity recognition from accelerometer data on a mobile phone. Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, pages 796–799, 2009.
- [3] W. Wu, S. Dasgupta, E.E. Ramirez, C. Peterson, and G.J. Norman. Classification accuracies of physical activities using smartphone motion sensors. Journal of Medical Internet Research, 14(5), 2012.
- [4] M.F.A. bin Abdullah, A.F.P. Negara, M.S. Sayeed, D.J. Choi, and K.S. Muthu. Classification algorithms in human activity recognition using smartphones. International Journal of Computer and Information Engineering, 6:77–84, 2012.
- [5] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine, 10(1):156–167, 2006.
- [6] V.W. Zheng, D.H. Hu, and Q. Yang. Cross-domain activity recognition. In Proceedings of the 11th international conference on Ubiquitous computing, pages 61–70, 2009.
- [7] L. Bao and S.S. Intille. Activity recognition from user-annotated acceleration data. In Pervasive Computing, pages 1–17. 2004.
- [8] U. Maurer, A. Smailagic, D.P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In IEEE International Workshop on Wearable and Implantable Body Sensor Networks, pages 4–pp, 2006.
- [9] L. Atallah, B. Lo, R. King, and Guang-Zhong Yang. Sensor placement for activity detection using wearable accelerometers. In International Conference on Body Sensor Networks, pages 24 –29, june 2010.
- [10] A. Rodríguez-Íguez-Molinero, D. Pérez-Martínez, A. Samá, P. Sanz, M. Calopa, C. Galvez, C. Perez-Lopez, J. Romagosa, and A. Catalá. Detection of gait parameters, bradykinesia and falls in patients with parkinson’s disease by using a unique triaxial accelerometer. World Parkinson Congress, 2007.
- [11] L. Bao and S.S. Intille. Activity recognition from user-annotated acceleration data. In Pervasive Computing, pages 1–17. 2004.
- [12] O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. IEEE Communications Surveys Tutorials, PP(99):1 –18, 2012.
- [13] H. Zhenyu and J. Lianwen. Activity recognition from acceleration data based on discrete cosine transform and svm. In IEEE International Conference on Systems, Man and Cybernetics, pages 5041 –5044, oct. 2009.

- [14] J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In IEEE International Conference on Systems, Man, and Cybernetics, volume 2, pages 747–752, 2001
- [15] M. Stikic, K. van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In IEEE International Symposium on Wearable Computers, pages 81–88, 28 2008-oct. 1 2008.
- [16] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In Proceedings of the National Conference on Artificial Intelligence, volume 20, page 21, 2005.
- [17] Miikka Ermes, Juha Parkka, and Luc Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 4451–4454, 2008.
- [18] B. Coley, B. Najafi, A.P. Ionescu, and K. Aminian. Stair climbing detection during daily physical activity using a miniature gyroscope. *Gait & posture*, 22(4):287–294, 2005
- [19] A.M. Khan, Y.-K. Lee, S.Y. Lee, and T.-S. Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In International Conference on Future Information Technology, pages 1–6, 2010
- [20] E.M. Tapia, S.S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In IEEE International Symposium on Wearable Computers, pages 37–40, 2007.
- [21] C. Zhu and W. Sheng. Human daily activity recognition in robot-assisted living using multi-sensor fusion. In IEEE International Conference on Robotics and Automation, pages 2154–2159, 2009.
- [22] A. Mannini and A.M. Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.
- [23] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997
- [24] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In Proceedings of the International Workshop of Ambient Assisted Living, 2012.
- [25] D. Anguita, G. Alessandro, O. Luca & P. Xavier, Reyes-Ortiz, J.L. (2013). Energy efficient smartphone-based activity recognition using fixed-point arithmetic. *Journal of Universal Computer Science*. 19. 1295-1314.
- [26] Reyes-Ortiz, Jorge Luis, et al. "Human Activity and Motion Disorder Recognition: towards smarter Interactive Cognitive Environments." *ESANN*. 2013.
- [27] Bao, L., Intille, S.: "Activity recognition from user-annotated acceleration data"; *Pervasive Computing*; 2004.

- [28] M. Espinilla, J. Medina, A. Salguero, N. Irvine, M. Donnelly, I. Cleland and C. Nugent. Human activity recognition from the acceleration data of a wearable device. Which features are more relevant by activities?. Proceedings. 2.1242. 2018.
- [29] Anguita, Davide, et al. "A public domain dataset for human activity recognition using smartphones." *Esann*. Vol. 3. 2013.
- [30] <https://www.thecapture.club>, Capture, a mobile app for tracking, reducing and removing one's CO2 footprint
- [31] Reyes-Ortiz, J.L., Ghio, A., Parra, X., Anguita, D., Cabestany, J. and Catala, A., 2013, April. Human Activity and Motion Disorder Recognition: towards smarter Interactive Cognitive Environments. In *ESANN*.
- [32] J. Biancat, C. Brighenti and A. Brighenti. Review of transportation mode detection techniques. ICST Transactions on Ambient Systems. 1. e7. 10.4108/amsys.1.4.e7. 2014.
- [33] A. Efthymiou, E. N. Barmounakis, D. Efthymiou and E. Vlahogianni. Transportation mode detection from low-power smartphone sensors using tree-based ensembles. J. Big Data Anal. Transp. 1, pp. 57–69, 2019.
- [34] Oroeco Inc. Oroeco. Mobile app. Available from: <https://www.oroeco.com/>, 2018.
- [35] A. Parande. GoGreen:Carbon Tracker. Mobile app. Version 1.2. Apple App Store, <https://apps.apple.com/us/app/gogreen-carbon-tracker/id1267386771>, 2018.
- [36] J. Kuc, Carbon Footprint Calculator. Mobile app. Version 1.1,Google Play Store, <https://play.google.com/store/apps/details?id=com.carbonfootprintcalc.kuc&hl=en>, 2013.
- [37] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten. The weka data mining software: An update. ACM SIGKDD Explor. Newslett. 11, 10–18.2009.
- [38] P. Domingos, M. Pazzani, On the optimality of the simple bayesian classifier under zero-one loss. Mach. Learn. 1997, 29, 103–137.
- [39] T. Cover, P. Hart, Nearest neighbor pattern classification. IEEE Trans. Inf. Theory 1967, 13, 21–27, doi: 10.1109/TIT.1967.1053964.
- [40] The J48 (C4.5) Algorithm (https://en.wikipedia.org/wiki/C4.5_algorithm)
- [41] The PART Algorithm
(<https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/PART.html>)
- [42] G. Mantellos, T. P. Exarhos, E. Christopoulou, Human activity and transportation mode recognition using smartphone sensors