



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

*Αναγνώριση Χειρονομιών για
εφαρμογές διάχυτου υπολογισμού με
χρήση έξυπνου ρολογιού*

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΥ ΕΜΜΑΝΟΥΗΛ

Επιβλέπων Καθηγητής

Χρήστος Γκουμόπουλος

Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή Εξεταστική Επιτροπή

.....
Καθηγητής 1

.....
Καθηγητής 2

.....
Καθηγητής 3

Σάμος, Οκτώβριος 2019

ΠΕΡΙΛΗΨΗ

Η αναγνώριση χειρονομιών του καρπού είναι μία πρόκληση για την αλληλεπίδραση ανθρώπου-υπολογιστή στο πεδίο του διάχυτου υπολογισμού που μπορεί να διευκολύνει τις ζωές τόσο των επαγγελματιών που ασκούν χειρωνακτική εργασία όσο και ατόμων με κινητικά προβλήματα. Στην παρούσα διπλωματική εργασία σχεδιάστηκε, αναπτύχθηκε και αξιολογήθηκε ένα σύστημα που μπορεί να αναγνωρίσει διακριτές χειρονομίες χρησιμοποιώντας το επιταχυνσιόμετρο ενός έξυπνου ρολογιού και μία εφαρμογή σε μια συσκευή κινητού υπολογισμού. Το έξυπνο ρολόι μεταδίδει στη συσκευή κινητού υπολογισμού, μέσω ασύρματης τεχνολογίας Bluetooth, τις τιμές που ανιχνεύει το επιταχυνσιόμετρο κατά τη διάρκεια εκτέλεσης της χειρονομίας. Στη συνέχεια αναλαμβάνει την αναγνώριση της χειρονομίας ο εκάστοτε αλγόριθμος που επιλέγει ο χρήστης μέσω της διεπαφής της εφαρμογής στη συσκευή. Στη χρήση του αλγόριθμου FFT, οι τιμές του επιταχυνσιόμετρου (χρονοσειρά) περνούν από μετασχηματισμό Fourier, υπολογίζεται το μέτρο τους και η χειρονομία κατηγοριοποιείται με μηχανή διανυσμάτων υποστήριξης. Στη χρήση του αλγόριθμου SAX η χρονοσειρά περνάει από τη μέθοδο μείωσης διαστάσεων PAA και στη συνέχεια κάθε τιμή της μειωμένης χρονοσειράς αντιστοιχίζεται με ένα σύμβολο ανάλογα με το εύρος τιμών στο οποίο ανήκει. Η χειρονομία αναγνωρίζεται βρίσκοντας τη μικρότερη απόσταση ανάμεσα στη συμβολοσειρά που προκύπτει και τις αποθηκευμένες στο αρχείο εκπαίδευσης. Το χαμηλό υπολογιστικό κόστος των παραπάνω αλγορίθμων επιτρέπουν τη χρήση του συστήματος με αναγνώριση σε πραγματικό χρόνο. Το σύστημα αξιολογήθηκε με δείγμα 5 ατόμων και μπορεί να αναγνωρίζει 5 χειρονομίες. Το μεγαλύτερο ποσοστό επιτυχίας που επιτεύχθηκε με το user-dependent evaluation ήταν 100% με τον αλγόριθμο SAX και με το user-independent 96% με τον SAX.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:

Επιταχυνσιόμετρο, διάχυτος υπολογισμός, έξυπνο ρολόι, συσκευή κινητού υπολογισμού, Bluetooth, αναγνώριση χειρονομίας, χρονοσειρά, αλγόριθμος

ABSTRACT

Wrist gesture recognition is a challenge for human-computer interaction in the field of ubiquitous computing that can improve the lives of both manual labor workers and people with mobility problems. During this thesis we developed a system that can recognize discrete gestures using the accelerometer of a smart watch and an application on a smart phone. While a gesture is performed, the smart watch transmits accelerometer data to the smart phone via Bluetooth wireless technology. Then the user chooses the preferred gesture recognition algorithm through the application interface of the device. By choosing the FFT algorithm, the accelerometer values of the time series go through a Fourier transformation, their absolute values are then calculated and the gesture is classified with a trained SVM. Using SAX algorithm, the time series goes through the PAA dimension reduction method, and then each value of the reduced time series is assigned a symbol depending on the range of values to which it belongs. The gesture is recognized by finding the shortest distance between the resulting string and those stored in the training file. The low computational cost of the above algorithms allows real-time recognition of the system. The system was evaluated with a sample of 5 people and can recognize 5 gestures. The best success rate achieved was 100% on user-dependent evaluation with SAX and 96% on user-independent evaluation using SAX.

KEYWORDS:

Accelerometer, ubiquitous computing, smart watch, smart phone, Bluetooth, gesture recognition, time series, algorithm, transform

ΠΕΡΙΕΧΟΜΕΝΑ

<u>ΠΕΡΙΛΗΨΗ</u>	2
<u>ABSTRACT</u>	3
<u>ΠΕΡΙΕΧΟΜΕΝΑ</u>	4
<u>ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ</u>	6
1.1 <u>Αντικείμενο Εργασίας</u>	6
1.2 <u>Σκοπιμότητα και Οφέλη</u>	7
1.3 <u>Δομή Εργασίας</u>	8
<u>ΚΕΦΑΛΑΙΟ 2: ΜΕΛΕΤΗ ΥΦΙΣΤΑΜΕΝΗΣ ΚΑΤΑΣΤΑΣΗΣ</u>	10
2.1 <u>Σχετικές Μελέτες και Εργασίες</u>	10
2.2 <u>Διαφοροποίηση Παρούσας Προσέγγισης</u>	13
<u>ΚΕΦΑΛΑΙΟ 3: ΚΑΤΑΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ</u>	14
<u>ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ</u>	16
4.1 <u>Διαδικασία Αναγνώρισης Χειρονομιών</u>	16
4.2 <u>Λογισμικό</u>	18
4.3 <u>Δομή Αρχείων Αποθήκευσης</u>	22
<u>ΚΕΦΑΛΑΙΟ 5: ΠΡΟΕΤΟΙΜΑΣΙΑ ΚΑΙ ΧΡΗΣΗ ΕΡΓΑΛΕΙΩΝ ΑΝΑΠΤΥΞΗΣ</u>	26
5.1 <u>Οι Διακριτές Χειρονομίες</u>	26
5.2 <u>Οι συσκευές που Χρησιμοποιήθηκαν</u>	28
5.3 <u>Εργαλεία Ανάπτυξης</u>	29
5.3.1 <u>Android Studio</u>	29
5.3.2 <u>Notepad++</u>	30
5.3.3 <u>OpenOffice Calc</u>	31
5.3.4 <u>RStudio</u>	32
5.4 <u>Τεχνολογίες και Βιβλιοθήκες</u>	33
5.4.1 <u>Java</u>	33
5.4.2 <u>XML</u>	33
5.4.3 <u>SPMF</u>	34
5.4.4 <u>LibSvm</u>	34
5.4.5 <u>OpenCSV</u>	34
<u>ΚΕΦΑΛΑΙΟ 6: ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ</u>	35
6.1 <u>Διάρθρωση Συστήματος</u>	35

6.1.1 Ανάπτυξη Εφαρμογής Έξυπνου Ρολογιού	36
6.1.2 Ανάπτυξη Εφαρμογής Συσκευής Κινητού Υπολογισμού	40
6.2 Γραφικό Περιβάλλον και Πλοήγηση	45
6.3 Εξαγωγή Δεδομένων	50
6.4 Αλγόριθμος FFT	52
6.4.1 Επεξεργασία Δεδομένων	52
6.4.2 Προσπέλαση Δεδομένων	52
6.4.3 Κατηγοριοποίηση Χειρονομίας	53
6.5 Αλγόριθμος SAX	54
6.5.1 Επεξεργασία Δεδομένων	54
6.5.2 Προσπέλαση Δεδομένων	55
6.5.3 Κατηγοριοποίηση Χειρονομίας	56
ΚΕΦΑΛΑΙΟ 7: ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ	58
7.1 Κατηγοριοποίηση Χειρονομιών	58
7.1.2 Κατηγοριοποίηση με τον FFT	59
7.1.3 Κατηγοριοποίηση με τον SAX	61
7.1.3.1 Κατηγοριοποίηση με Καθολικό Training Set	61
7.1.3.2 Κατηγοριοποίηση με Προσωποκεντρικό Training Set	63
7.2 Συγκεντρωτικά Αποτελέσματα	65
ΚΕΦΑΛΑΙΟ 8: ΕΠΙΛΟΓΟΣ	67
8.1 Συμπεράσματα	67
8.2 Μελλοντικές Βελτιώσεις	68
ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ	70
ΠΑΡΑΡΤΗΜΑ Α – Οι Διακριτές Χειρονομίες	73
ΠΑΡΑΡΤΗΜΑ Β – Οδηγίες για τη Δημιουργία Project με Wear και Mobile Module στο Android Studio	76

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο Εργασίας

Οι αποτελεσματικότερες τεχνολογίες είναι αυτές οι οποίες εξαφανίζονται, με την έννοια ότι ενσωματώνονται με τέτοιο τρόπο στο περιβάλλον τους ώστε καθίστανται αδιαχώριστες από αυτό [1]. Το ανωτέρω συνιστά το όραμα του διάχυτου (pervasive) ή πανταχού παρόντος (ubiquitous) υπολογισμού. Οι υπολογιστικές συσκευές δεν αποτελούν πλέον διακριτό τμήμα της πραγματικότητας, αλλά εντάσσονται σε άλλα υπάρχοντα τμήματα, δημιουργώντας ένα πλαίσιο «ενσωματωμένης εικονικότητας». Στην ουσία, ο διάχυτος υπολογισμός αποτελεί ένα διαφορετικό παράδειγμα στον τομέα της επιστήμης των υπολογιστών, στου οποίου το επίκεντρο είναι ο χρήστης και όχι ο υπολογιστής [2]. Η συνεχής βελτίωση της ακρίβειας των αισθητήρων και το μικρό κόστος κατασκευής τους έχει καταστήσει δεδομένη την ενσωμάτωσή τους στην πλειοψηφία των έξυπνων συσκευών, που κατασκευάζονται τα τελευταία χρόνια (ρολόγια, κινητά τηλέφωνα, τάμπλετ κ.α.). Η βοήθεια που προσφέρουν οι αλγόριθμοι επεξεργασίας σήματος σε συνδυασμό με τις μεθόδους μηχανικής μάθησης κάνουν εφικτή την ανάπτυξη συστημάτων που αναγνωρίζουν τις χειρονομίες των χεριών. Με αυτό τον τρόπο κάθε χρήστης συνδέεται άμεσα ή έμμεσα με πολλές υπολογιστικές συσκευές με σχεδόν ανεπαίσθητο τρόπο, οι οποίες τον βοηθούν στην εκτέλεση των δραστηριοτήτων του [2].

Τα έξυπνα ρολόγια είναι φορητοί υπολογιστές στη μορφή ενός ρολογιού καρπού που μπορούν να εκτελούν απλές διαδικασίες όπως εμφάνιση της ώρας ή ειδοποίηση εισερχομένων κλήσεων και μηνυμάτων από το έξυπνο κινητό με το οποίο βρίσκονται σε σύζευξη, αλλά και πολύπλοκες όπως αναπαραγωγή πολυμέσων, οδηγίες διαδρομής μέσω GPS, καταγραφή βιομετρικών τιμών του χρήστη. Τα περισσότερα διαθέτουν αισθητήρες όπως βαρόμετρο, επιταχυνσιόμετρο, γυροσκόπιο, πυξίδα, δέκτες GPS [3]. Συγκεκριμένα το Huawei Watch 2 Sport που χρησιμοποιεί το λειτουργικό σύστημα Wear OS 2.0 της Google (πρώην Android Wear) διαθέτει επιταχυνσιόμετρο τριών αξόνων που μπορεί να καταγράφει τις κινήσεις του χεριού του χρήστη [4]. Αυτό όπως και άλλα smartwatch έχουν χρησιμοποιηθεί σε αρκετά παραδείγματα όπως τον τηλεχειρισμό ρομπότ [5], την εισαγωγή κειμένου χωρίς δακτυλογράφηση [6], τον έλεγχο μιας διεπαφής με τις χειρονομίες του καρπού [7].

Το αντικείμενο της διπλωματικής εργασίας είναι η ανάπτυξη και η αξιολόγηση ενός συστήματος αναγνώρισης χειρονομιών που θα βοηθήσει, στον τηλεχειρισμό συσκευών, άτομα που ασκούν χειρωνακτική εργασία ή έχουν κινητικά προβλήματα. Το σύστημα απαιτεί τη χρήση ενός έξυπνου ρολογιού με επιταχυνσιόμετρο τριών αξόνων και συνδεσιμότητα Bluetooth που χρησιμοποιεί το λειτουργικό σύστημα Wear OS με API level 28 και άνω (Wear OS 2.0), όπως επίσης και μια συσκευή κινητού υπολογισμού με λειτουργικό σύστημα Android με API level 28

και άνω (Android 8.0 Oreo) καθώς αυτές είναι οι ελάχιστες απαιτήσεις με τις οποίες αναπτύχθηκαν οι εφαρμογές.

Στην παρούσα κατάσταση, το σύστημα είναι ικανό να αναγνωρίζει 5 χειρονομίες. Συγκεκριμένα:

1. hh → hand horizontal
2. hu → hand up
3. hud → hand up direct
4. hh2 → hand horizontal twice
5. hu2 → hand up twice

Χρησιμοποιώντας ένα training set των 250 τιμών που συντάχθηκε με τις μετρήσεις 5 διαφορετικών ατόμων και τον αλγόριθμο αναγνώρισης FFT επιτυγχάνεται ποσοστό επιτυχίας 96% σε προσωποκεντρικό training set και σε user- independent 94% ενώ ο αλγόριθμος έχει ποσοστά επιτυχίας SAX 100% και 96% αντίστοιχα.

1.2 Σκοπιμότητα και Οφέλη

Η Μαρία είναι οδοντίατρος και εξετάζει ασθενείς στο ιατρείο της. Κατά τη διάρκεια της εξέτασης προκύπτει η ανάγκη να ρυθμίσει τον κλιματισμό του ιατρείου, να χειριστεί τον προσωπικό της υπολογιστή για να ελέγξει τις ακτινογραφίες του εκάστοτε ασθενή και να ανοίξει την πόρτα της πολυκατοικίας και του ιατρείου ώστε να περάσουν οι επόμενοι ασθενείς στην αίθουσα αναμονής. Θα ήθελε για αυτές τις ρυθμίσεις να μην απαιτείται να χρησιμοποιήσει τα δάχτυλα της καθώς το τηλεχειριστήριο του κλιματιστικού, το ποντίκι του υπολογιστή και τα κουμπιά που ανοίγουν τις πόρτες είναι καλυμμένα με μικρόβια που μπορεί να μολύνουν τα ιατρικά εργαλεία και κατ' επέκταση τον ασθενή. Για το λόγο αυτό χρησιμοποιεί ιατρικά αποστειρωμένα γάντια τα οποία αλλάζει μετά τη χρήση της κάθε συσκευής με αποτέλεσμα να χάνει χρόνο και να ξοδεύει χρήματα στην αγορά γαντιών. Τη λύση στη σπατάλη γαντιών και χρόνου μπορεί να δώσει ένα έξυπνο ρολόι που θα αναγνωρίσει απλές φυσικές κινήσεις που η ίδια έχει ορίσει. Θα χρειαστεί μια κίνηση για να αλλάξει από το ένα αντικείμενο χειρισμού στο άλλο, κλιματισμός, υπολογιστής, πόρτες. Για τη λειτουργία αυτή μπορεί να χρησιμοποιηθεί η χειρονομία hud κατά την οποία γίνεται αλλαγή στον προσανατολισμό του χεριού. Ο αριθμός των αντικειμένων είναι μικρός και αρκεί η κυκλική εναλλαγή. Θα χρειαστεί επίσης, μια κίνηση για να αυξήσει βηματικά τη ρύθμιση του κλιματισμού, και μια για να τη μειώσει βηματικά όπως επίσης μια κίνηση που θα φέρει το μέγιστο της ρύθμισης και μια για το ελάχιστο της ρύθμισης. Για τη βηματική αύξηση μπορεί να χρησιμοποιηθεί η χειρονομία hu κατά την οποία το χέρι σηκώνεται μέχρι το ύψος των ματιών και για τη βηματική μείωση η χειρονομία hh, ενώ για τη

μέγιστη και την ελάχιστη ρύθμιση μπορούν να χρησιμοποιηθούν οι χειρονομίες hu2 και hh2 αντίστοιχα. Τις ίδιες χειρονομίες θα χρησιμοποιήσει και για το χειρισμό των διαφανειών με τις ακτινογραφίες στον υπολογιστή. Για την επόμενη διαφάνεια θα χρησιμοποιηθεί η χειρονομία hh και για την προηγούμενη η hu ενώ για να διαλέξει την τελευταία διαφάνεια μπορεί να χρησιμοποιηθεί η χειρονομία hh2 και για την πρώτη η hu2. Όσον αφορά τις πόρτες, η κίνηση hh μπορεί να αντιστοιχιστεί για το άνοιγμα της πόρτας της πολυκατοικίας με μικρή διάρκεια στο πάτημα του κουμπιού και η hh2 με μεγάλη διάρκεια. Αντίστοιχα για την πόρτα του ιατρείου μπορούν να χρησιμοποιηθούν και οι χειρονομίες hu και hu2. Τη σύνδεση με τις αντίστοιχες συσκευές και το χειρισμό τους, θα αναλάβει η συσκευή κινητού υπολογισμού, που είναι συνδεδεμένη με το έξυπνο ρολόι, αφού αναγνωρίσει την χειρονομία.

Ο Γιώργος κινείται με το αναπηρικό καροτσάκι μετά το ατύχημα, με αποτέλεσμα να μη φτάνει με ευκολία τους διακόπτες του φωτός στο σπίτι του. Στην προκειμένη, η χειρονομία hud μπορεί να χρησιμοποιηθεί για την εναλλαγή των δωματίων ενώ οι hu και hh για να τα ανάψουν και τα να σβήσουν αντίστοιχα. Στην περίπτωση που έχει χάσει κάτι και χρειάζεται όλα τα φώτα ανοιχτά για να το βρει θα τον βοηθήσει η χειρονομία hu2 ενώ αν ξαπλώσει για ύπνο ξεχνώντας κάποια φώτα ανοιχτά στα άλλα δωμάτια μπορεί να τα κλείσει όλα μαζί με τη χειρονομία hh2.

Επομένως, τα οφέλη που σκοπεύουμε να επιτύχουμε με αυτή την εφαρμογή είναι η βελτίωση της καθημερινότητας των ανθρώπων που χρησιμοποιούν τα χέρια και τα δάχτυλά τους στη δουλειά ή τις ασχολίες τους και των ανθρώπων που έχουν κινητικά προβλήματα. Ανάλογα με το σενάριο θα πρέπει να γίνει ξεχωριστή εκπαίδευση της εφαρμογής ώστε να ρυθμιστεί σωστά και να λειτουργεί με τη μέγιστη δυνατή ακρίβεια.

1.3 Δομή Εργασίας

Η παρούσα διπλωματική εργασία είναι χωρισμένη σε οκτώ κεφάλαια, τα οποία είναι τα παρακάτω:

◆ Κεφάλαιο 1: Εισαγωγή

Στο κεφάλαιο αυτό ο αναγνώστης ενημερώνεται για το αντικείμενο, τη σκοπιμότητα και τα οφέλη αυτής της εργασίας.

◆ Κεφάλαιο 2: Μελέτη Υφιστάμενης Κατάστασης

Στο κεφάλαιο αυτό μελετάται η υφιστάμενη κατάσταση στον τομέα της αναγνώρισης χειρονομιών, οι μελέτες που υπάρχουν σχετικά με το αντικείμενο της εργασίας και επιλέγεται η μεθοδολογία ανάπτυξης του συστήματος.

◆ Κεφάλαιο 3: Καταγραφή και Ανάλυση Απαιτήσεων

Στο κεφάλαιο αυτό καταγράφονται οι απαιτήσεις του συστήματος, βρίσκοντας ομοιότητες και διαφορές, αναλύοντας τις σχετικές εργασίες στο Κεφάλαιο 2.

◆ Κεφάλαιο 4: Σχεδιασμός Συστήματος

Στο κεφάλαιο αυτό σχεδιάζεται το σύστημα με τις εφαρμογές σε smart watch και smart phone επισημαίνοντας τους λόγους που προτιμήθηκε αυτή η προσέγγιση του ζητήματος της αναγνώρισης χειρονομιών.

◆ Κεφάλαιο 5: Προετοιμασία και Χρήση Εργαλείων Ανάπτυξης

Στο κεφάλαιο αυτό γίνεται η απαραίτητη προετοιμασία για την έναρξη της ανάπτυξης των εφαρμογών, ελέγχονται οι συσκευές ώστε να ικανοποιούνται όλες οι προϋποθέσεις για την επιτυχή λειτουργία του συστήματος και αναφέρονται όλες οι εφαρμογές, οι βιβλιοθήκες και τα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη του συστήματος.

◆ Κεφάλαιο 6: Υλοποίηση Συστήματος

Στο κεφάλαιο αυτό αναφέρεται αναλυτικά η υλοποίηση του συστήματος όσον αφορά το περιβάλλον ανάπτυξης, τα βασικά στοιχεία του, τον τρόπο αποθήκευσης, μετασχηματισμού και προσπέλασης των δεδομένων από το επιταχυνσιόμετρο καθώς και τους αλγόριθμους αναγνώρισης. Περιλαμβάνονται εικόνες των χειρονομιών και τμήματα του κώδικα που αναπτύχθηκε.

◆ Κεφάλαιο 7: Αξιολόγηση Συστήματος

Στο κεφάλαιο αυτό αξιολογούνται οι διαφορετικές μέθοδοι κατηγοριοποίησης βάσει δύο διαφορετικών training set με τη βοήθεια 5 ατόμων και συγκρίνεται η ακρίβειά τους ώστε να κριθεί κάποιος πιο αποδοτικός.

◆ Κεφάλαιο 8: Επίλογος

Στο κεφάλαιο αυτό καταγράφονται τα συμπεράσματα, οι αδυναμίες του συστήματος και οι μελλοντικές βελτιώσεις που επιδέχεται για την αύξηση της απόδοσης ή την ευκολότερη χρήση του.

ΚΕΦΑΛΑΙΟ 2: ΜΕΛΕΤΗ ΥΦΙΣΤΑΜΕΝΗΣ ΚΑΤΑΣΤΑΣΗΣ

Για το σωστό σχεδιασμό και την ανάπτυξη του συστήματος έγινε εκτενής έρευνα για παρόμοιες μελέτες που έχουν εκπονηθεί στο παρελθόν και έγινε ανάλυση της μεθοδολογίας και των συμπερασμάτων που προέκυψαν όπως εξετάστηκαν και τα προβλήματα και οι πιθανοί τρόποι βελτίωσης των μεθόδων.

Όλες οι πηγές προέρχονται από το Google Scholar, το εργαλείο αναζήτησης της Google που εμφανίζει αποτελέσματα από έρευνες και μελέτες που έχουν δημοσιευθεί στο διαδίκτυο.

2.1 Σχετικές Μελέτες και Εργασίες

ACCELEROMETER GESTURE RECOGNITION – Michael Xie, David Pan – University of Stanford [8]

Σε αυτή τη μελέτη, οι Xie και Pan, ανέπτυξαν ένα SVM υψηλής ακρίβειας που απαιτεί ένα μόνο δείγμα εκπαίδευσης ανά κλάση. Αυτό το πέτυχαν χρησιμοποιώντας τον αλγόριθμο Dynamic-Threshold Truncation ως φίλτρο κατά των δεδομένων που αποστέλλονται από το επιταχυνσιόμετρο ενός κινητού τηλεφώνου διαρκώς μέχρι να εντοπιστούν τα δεδομένα μιας χειρονομίας για αναγνώριση και μετασχηματισμό Fourier στα δεδομένα κάθε άξονα του επιταχυνσιόμετρου για την κατηγοριοποίηση. Η ακρίβεια της μεθόδου τους κατά την αξιολόγησή της με ένα σετ 31 χειρονομιών που ανήκουν σε 5 διαφορετικές κλάσεις κυμαίνεται από 96% με ένα δείγμα εκπαίδευσης και με 5 δείγματα 98% χρησιμοποιώντας κρυφά Μαρκοβιανά μοντέλα.

ACCELEROMETER BASED GESTURE RECOGNITION USING FUSION FEATURES AND SVM – Zhenyu He – Computer Center, Jinan University, Guangzhou, China [9]

Ο He πρότεινε έναν υβριδικό τρόπο αναγνώρισης χειρονομιών για την επικοινωνία ανθρώπου υπολογιστή με το επιταχυνσιόμετρο ενός κινητού τηλεφώνου χρησιμοποιώντας τα τμήματα της χρονοσειράς των δεδομένων που έχουν υψηλή ενέργεια σε μικρό χρονικό διάστημα και των δεδομένων που έχουν περάσει από μετασχηματισμό Fourier και Wavelet Packet Decomposition. Η μέθοδος αυτή αναγνώρισε δεκαεφτά σύνθετες χειρονομίες, με δεδομένα χειρονομιών από 67 άτομα, με ποσοστό επιτυχίας 89.89%.

THE GESTURE WATCH: A WIRELESS CONTACT-FREE GESTURE BASED WRIST INTERFACE – *Jungsoo Kim, Jiasheng He, Kent Lyons, Thad Starner - College of Computing and GVI Center Georgia Institute of Technology Atlanta, GA 30332-0280 USA [10]*

Σε αυτή την ερευνητική προσπάθεια ο Kim κ.ά. πρότειναν την ενσωμάτωση πέντε υπέρυθρων αισθητήρων εγγύτητας γύρω από την οθόνη ενός έξυπνου ρολογιού για την ανέπαφη αναγνώριση χειρονομιών όταν ο χρήστης περνάει το χέρι του πάνω από την οθόνη. Ένα πιθανό σενάριο χρήσης της ιδέας είναι ο έλεγχος ενός mp3 player ενώ ο χρήστης είναι στο δρόμο. Έκαναν χρήση των κρυφών Μαρκοβιανών μοντέλων για την αναγνώριση των πέντε χειρονομιών, που εκτέλεσαν 10 διαφορετικά άτομα και οι δοκιμές εν κινήσει και σε περιβάλλον με ακραίες συνθήκες φωτισμού έφεραν ποσοστό επιτυχίας 95.5%.

uWAVE: ACCELEROMETER-BASED PERSONALIZED GESTURE RECOGNITION AND ITS APPLICATIONS – *Jiayang Liu, Zhen Wang and Lin Zhong - Department. Of Electrical Computer Engineering Rice University, Houston TX 77005, Jehan Wickramasuriya and Venu Vasudevan - Pervasive Platforms & Architectures Lab Applications & Software Research Center, Motorola Labs [11]*

Αυτή η μελέτη πέτυχε μεγάλη ακρίβεια στην αναγνώριση χειρονομιών επιτρέποντας στους χρήστες τη χρήση προσωπικών χειρονομιών με συσκευές που διαθέτουν επιταχυνσιόμετρο τριών αξόνων. Για την εκπαίδευση του συστήματος λήφθηκαν 4000 δείγματα χειρονομιών από 8 άτομα. Με τη βοήθεια του αλγόριθμου Dynamic Time Warping για την εύρεση ομοιοτήτων μεταξύ των χρονοσειρών από τα δεδομένα του επιταχυνσιόμετρου, τον κβαντισμό των δεδομένων για τη μείωση του υπολογιστικού κόστους και του θορύβου του σήματος και την προσαρμογή του λεξιλογίου για πιο ακριβή αναγνώριση στο πέρασμα του χρόνου το ποσοστό ακρίβειας του αλγορίθμου έφτασε το 98.6% με μόλις ένα δείγμα εκπαίδευσης ανά κλάση.

GESTURE RECOGNITION WITH A 3-D ACCELEROMETER – *Jianhui Wu, Gang Pan, Daqing Zhang, Guande Qi, Shijian Li – Department of Computer Science, Zhejiang University, Hangzhou, 310027, China, Handicom Lab, Institut TELECOM SudParis, France [12]*

Η προσέγγιση για την αναγνώριση χειρονομιών που ακολούθησαν ο Wu κ.ά. σε αυτή τη μελέτη βασίζεται σε ένα Frame-based Descriptor όπου διαβάζει τα εισερχόμενα δεδομένα από ένα επιταχυνσιόμετρο τριών αξόνων και ανιχνεύει τη χρήσιμη πληροφορία της χειρονομίας και ένα multiclass SVM για την κατηγοριοποίηση της κάθε χειρονομίας. Με ένα σετ δεδομένων χειρονομιών πλήθους 3360 που αποτελείται από 12 διακριτές χειρονομίες επιτυγχάνει ποσοστό ακρίβειας 95.21% στην αξιολόγηση από άτομα που συμμετείχαν στην εκπαίδευση του data set και 89.29% από άτομα που δε συμμετείχαν.

A FRAMEWORK FOR HAND GESTURE RECOGNITION BASED ON ACCELEROMETER AND EMG SENSORS – Xu Zhang Chen, Associate Member, IEEE, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang [13]

Σε αυτή τη μελέτη των Zhang Chen κ.ά. αξίζει να σημειωθεί πως για την αναγνώριση της έναρξης της χειρονομίας από το χρήστη, το σύστημα χρησιμοποιεί ηλεκτρομυογράφους στον πήχη του χεριού που ο χρήστης φοράει μαζί με το επιταχυνσιόμετρο τριών αξόνων. Τα δεδομένα από τους αισθητήρες συνδυάζονται και με χρήση κρυφών Μαρκοβιανών Μοντέλων φτάνουν το ποσοστό επιτυχίας στην αναγνώριση 18 χειρονομιών της κινεζικής νοηματικής γλώσσας στο 97.6% user-dependent και 90.2% user-independent.

GESTURE RECOGNITION WITH A Wii CONTROLER – Thomas Schlömer, Benjamin Poppinga, Niels Henze, Susanne Boll – University of Oldenburg, OFFIS Institute for Information Tecnology [14]

Η μελέτη αυτή εξετάζει τη χρήση του τηλεχειριστηρίου της κονσόλας Wii της Nintendo ως τηλεχειριστήριο για διάφορες ηλεκτρονικές συσκευές με χρήση πέντε διαφορετικών χειρονομιών χρησιμοποιώντας για την αναγνώριση των χειρονομιών φίλτρα για την εξαίρεση των δεδομένων όσο το σύστημα βρίσκεται σε αναμονή και για την ομαλοποίηση των δεδομένων της χειρονομίας. Στη συνέχεια οι χειρονομίες κατηγοριοποιούνται με χρήση κρυφών Μαρκοβιανών μοντέλων και το ποσοστό ακρίβειας που επιτεύχθηκε στην εκτέλεση των χειρονομιών από 6 άτομα έφτασε το 89.7%.

ACCELEROMETER-BASED GESTURE RECOGNITION VIA DYNAMIC-TIME WARPING, AFFINITY PROPAGATION, & COMPRESSIVE SENSING – Ahmad Akl and Shahrokh Valaee – Department of Electrical and Computer Engineering, University of Toronto [15]

Το σύστημα αναγνώρισης χειρονομιών που προτείνεται σε αυτή τη μελέτη κάνει χρήση των αλγορίθμων dynamic-time warping, affinity propagation για την εκπαίδευση του συστήματος με τα δεδομένα ενός επιταχυνσιόμετρου τριών αξόνων και του αλγόριθμου compressive sensing για την αναγνώριση της τροχιάς που ακολουθεί η κάθε χειρονομία. Δημιουργήθηκε ένα λεξιλόγιο από 18 διακριτές χειρονομίες και μια βάση δεδομένων που αποτελείται από 3700 εγγραφές αυτών των χειρονομιών που εκτέλεσαν 7 χρήστες. Η συνολική ακρίβεια του συστήματος μετά από αξιολόγηση από άτομα που συμμετείχαν στην εκπαίδευση του σετ και από άτομα που δε συμμετείχαν είναι και στις δυο περιπτώσεις 99,70% με την εκπαίδευση του συστήματος από τους νέους χρήστες εκτελώντας την κάθε χειρονομία 3 φορές.

2.2 Διαφοροποίηση Παρούσας Προσέγγισης

Στην πλειοψηφία των προσεγγίσεων των ερευνών που μελετήθηκαν χρησιμοποιούνται κρυφά Μαρκοβιανά μοντέλα και ο αλγόριθμος Dynamic Time Warping για την εκπαίδευση του συστήματος και την αναγνώριση των χειρονομιών ενώ οι υπόλοιπες κάνουν χρήση υπέρυθρων αισθητήρων εγγύτητας, ηλεκτρομυογράφων και φίλτρων για την ανίχνευση της σημαντικής πληροφορίας που εισέρχεται στο σύστημα από το επιταχυνσιόμετρο. Επίσης, οι περισσότερες προσεγγίσεις χρησιμοποιούν δεδομένα από επιταχυνσιόμετρο τριών αξόνων και χρησιμοποιούν για τη λήψη των δεδομένων συσκευές όπως κινητά τηλέφωνα ή τηλεχειριστήρια κονσολών και η αναγνώριση των χειρονομιών γίνεται σε συσκευές όπως Η/Υ, υπολογιστές τσέπης, κινητά τηλέφωνα.

Η προσέγγιση που παρουσιάζεται σε αυτή τη διπλωματική έχει κάποια κοινά χαρακτηριστικά με τις προσεγγίσεις που προαναφέρθηκαν καθώς χρησιμοποιεί ένα έξυπνο ρολόι που διαθέτει επιταχυνσιόμετρο τριών αξόνων και μια συσκευή κινητού υπολογισμού για την αναγνώριση των χειρονομιών. Χρησιμοποιεί το μετασχηματισμό δεδομένων με τη μέθοδο FFT όπως στα παραδείγματα [8] και [9] και για την κατηγοριοποίησή τους χρησιμοποιεί ένα multiclass SVM όπως στις προσεγγίσεις [8] και [12] με μόνη διαφορά το μικρότερο πλήθος των σημείων που αποτελείται το σήμα καθώς χρησιμοποιούνται μόνο 64 σημεία της χρονοσειράς μετά τον κβαντισμό των δεδομένων. Το σύστημα είναι σε θέση να αναγνωρίζει 5 διακριτές χειρονομίες όπως στις μελέτες [10] και [14]. Σε αντίθεση με τις υπόλοιπες μελέτες δεν έχει γίνει εφαρμογή φίλτρων στα εισερχόμενα δεδομένα καθώς ο χρήστης της εφαρμογής πρέπει να πιέσει ένα κουμπί στη διεπαφή της συσκευής κινητού υπολογισμού για να ξεκινήσει η καταγραφή των δεδομένων με αποτέλεσμα να μην υπάρχει θόρυβος ή ασήμαντη πληροφορία στο τελικό σήμα, εφόσον η αρχική και τελική θέση του χεριού του κάθε χρήστη είναι καθορισμένη στο χρονικό παράθυρο της καταγραφής. Όσον αφορά το δείγμα των δεδομένων που εκπαίδευσαν το σύστημα, αποτελείται από 250 εγγραφές χειρονομιών που το συνέταξαν 5 διαφορετικά άτομα, αντί για ένα δείγμα χιλιάδων εγγραφών όπως στην πλειοψηφία των προσεγγίσεων που περιεγράφηκαν, επιτυγχάνοντας εξίσου μεγάλο ποσοστό ακρίβειας. Μια ακόμη διαφοροποίηση από τις περισσότερες προσεγγίσεις είναι η δυνατότητα εκπαίδευσης του συστήματος από νέο χρήστη με την καταχώριση των νέων χειρονομιών που εκτελεί και η επιτυχής αναγνώρισή τους με το δεύτερο αλγόριθμο που υλοποιήθηκε. Ο δεύτερος αλγόριθμος που χρησιμοποιείται σε αυτό το σύστημα είναι μια ακόμη σημαντική διαφοροποίηση καθώς δε βρέθηκαν σχετικές αναφορές στη βιβλιογραφία για χρήση του σε παρόμοιο σύστημα. Πρόκειται για τον αλγόριθμο Symbolic Aggregate Approximation (SAX) του οποίου ο τρόπος λειτουργίας προσέφερε μεγάλο ποσοστό ακρίβειας ακόμα και σε ανεξάρτητους χρήστες κατά την αξιολόγηση του συστήματος. Εν κατακλείδι, ο σκοπός του συστήματος που αναπτύχθηκε, είναι κυρίως η σύγκριση της αποδοτικότητας των δύο αλγορίθμων που υλοποιήθηκαν και το ποσοστό ακρίβειας που επιτεύχθηκε, από τον καθένα, στην αναγνώριση των χειρονομιών με τη χρήση τους από άτομα που συμμετείχαν στη σύνταξη του αρχείου εκπαίδευσης και από ανεξάρτητους χρήστες.

ΚΕΦΑΛΑΙΟ 3: ΚΑΤΑΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Αναλύοντας τις μελέτες που προαναφέρθηκαν, προέκυψαν κάποια συμπεράσματα για την κατάλληλη προσέγγιση στο σχεδιασμό και την ανάπτυξη του συστήματος ώστε να επιτευχθεί μεγάλη ακρίβεια στην αναγνώριση χειρονομιών με συσκευές που είναι διαθέσιμες στους καταναλωτές.

Στη συνέχεια αναλύεται η καταγραφή απαιτήσεων του συστήματος, λαμβάνοντας υπόψιν τα δυνατά και αδύναμα σημεία των παραπάνω υλοποιήσεων ώστε να γίνει σωστός σχεδιασμός των εφαρμογών με στόχο την ευκολία στη χρήση, την ευκολία της ενσωμάτωσης ενός τέτοιου συστήματος στην καθημερινότητα, την πιθανή μελλοντική βελτίωση του συστήματος και τη συνολική ακρίβεια αναγνώρισης.

Προτού ξεκινήσει η υλοποίηση του συστήματος, αποφασίστηκε το λεξιλόγιο των χειρονομιών που αποτελείται από τις 5 χειρονομίες που αναφέρθηκαν στο κεφάλαιο 1.2 Σκοπιμότητα και Οφέλη στη σελίδα 8. Στη συνέχεια, μετά από επανειλημμένες εκτελέσεις των χειρονομιών, υπολογίστηκε το χρονικό παράθυρο που διαθέτει ο κάθε χρήστης για την εκτέλεση ολόκληρης της χειρονομίας, στα 5 δευτερόλεπτα.

Οι συσκευές που κρίθηκαν καταλληλότερες για την υλοποίηση ενός τέτοιου συστήματος είναι ένα smart watch για τη λήψη των δεδομένων και ένα smart phone για την αποθήκευση τους και την αναγνώριση. Για την αναγνώριση των χειρονομιών έπρεπε να υλοποιηθεί ένα σύστημα που θα έχει δύο λειτουργίες, εκπαίδευση και αναγνώριση. Στη λειτουργία της εκπαίδευσης η κάθε νέα χειρονομία προστίθεται σε ένα αρχείο αφού την ονομάσει ο χρήστης που την εκτέλεσε. Στη λειτουργία της αναγνώρισης, η καταγεγραμμένη χειρονομία συγκρίνεται με τις αποθηκευμένες με τον αντίστοιχο αλγόριθμο αναγνώρισης που επιλέγει ο χρήστης μέσω της διεπαφής και το αποτέλεσμα θα εμφανίζεται στην οθόνη του smart phone.

Ο σχεδιασμός των αρχείων για την αποθήκευση των δεδομένων προέκυψε από τις απαιτήσεις του κάθε αλγορίθμου σύμφωνα με τις βιβλιοθήκες που χρησιμοποιήθηκαν. Για την αποθήκευση των δεδομένων και την προσπέλασή τους, επιλέχθηκε η μορφή αρχείων .csv και .txt. Τέλος τα αρχεία θα περιέχουν τα «ωμά» και τα μετασχηματισμένα δεδομένα από τους αλγόριθμους για να αρχικοποιείται το σύστημα με την έναρξη της εφαρμογής ώστε να επιτευχθεί μεγαλύτερη ταχύτητα και να μπορεί να γίνει μελέτη των αποτελεσμάτων με σκοπό τη βελτίωση του ποσοστού ακριβείας του συστήματος.

Στη λίστα που ακολουθεί αναφέρονται οι λειτουργικές και μη-λειτουργικές απαιτήσεις του συστήματος:

- Μία συσκευή κινητού υπολογισμού με λειτουργικό Android [16] και μια συσκευή φορητής υπολογιστικής ικανότητας με λειτουργικό Wear OS[17].
- Η συνεπής εκτέλεση των χειρονομιών μέσα στο χρονικό παράθυρο που ορίστηκε.
- Ένα σύστημα που θα έχει λιτή διεπαφή με απλές και σαφείς οδηγίες χρήσης ώστε να είναι εύχρηστο ακόμα και σε άτομα που δεν έχουν μεγάλη εξοικείωση με παρόμοιες συσκευές.
- Ένα σύστημα που θα επιτυγχάνει μεγαλύτερο ή ίσο ποσοστό ακρίβειας στην αναγνώριση με μικρότερο όγκο δεδομένων και σε πραγματικό χρόνο.
- Η σωστή δομή των αρχείων για την αποθήκευση των χειρονομιών κατά τη διαδικασία της εκπαίδευσης και τη σύγκριση των εισερχόμενων δεδομένων της χειρονομίας με τις ήδη αποθηκευμένες κατά τη διαδικασία της αναγνώρισης

ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

4.1 Διαδικασία Αναγνώρισης Χειρονομιών

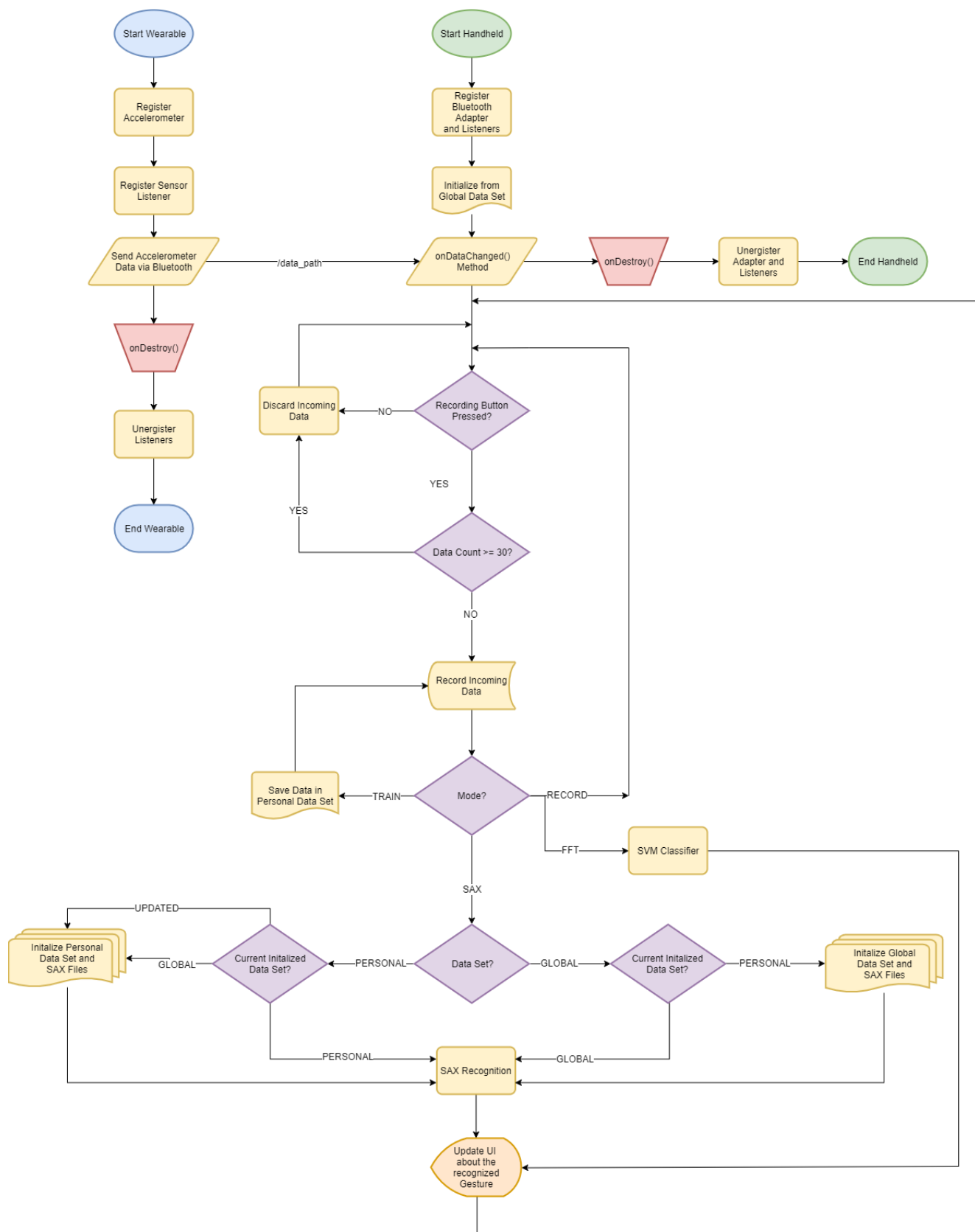
Μια πρόκληση στην ανάπτυξη ενός συστήματος αναγνώρισης χειρονομιών είναι η απόσπαση της σημαντικής πληροφορίας για τη χειρονομία μέσα από τη διαρκή ροή δεδομένων από το επιταχυνσιόμετρο του ρολογιού προς τη συσκευή που διαχειρίζεται την αναγνώριση, καθώς τα χέρια του χρήστη βρίσκονται διαρκώς σε κίνηση, ανάλογα με τις καθημερινές του ασχολίες. Στα πλαίσια αυτής της διπλωματικής εργασίας μελετάται μια απλή έκδοση του προβλήματος κατά την οποία ο χρήστης έχει στη διάθεσή του 5 διακριτές χειρονομίες που προσφέρει το σύστημα και μπορεί να τις εκτελέσει, είτε για αναγνώριση είτε για εκπαίδευση, με το πάτημα ενός κουμπιού στη διεπαφή της εφαρμογής. Αυτός ο τρόπος προσέγγισης επιλέχθηκε για να μειωθεί στο ελάχιστο ο θόρυβος του σήματος και να μην είναι απαραίτητη η χρήση φίλτρων και αλγορίθμων για την ανίχνευση της πληροφορίας της χειρονομίας στο συνεχές σήμα.

Το σήμα προέρχεται από το επιταχυνσιόμετρο τριών αξόνων που διαθέτει το έξυπνο ρολόι. Η συντριπτική πλειοψηφία των έξυπνων ρολογιών που κυκλοφορούν στην αγορά διαθέτει επιταχυνσιόμετρα τριών αξόνων με χαμηλό δείκτη θορύβου, πράγμα που τα καθιστά ως τις πιο κατάλληλες συσκευές για την ανάπτυξη ενός τέτοιου συστήματος, αφού επιτρέπουν την καταγραφή της θέσης του χεριού στο χώρο με υψηλή ακρίβεια.

Όλα τα πιθανά σενάρια χρήσης του συστήματος φαίνονται στο Διάγραμμα 4.1.1 [18]. Η εφαρμογή που έχει υλοποιηθεί στη συσκευή φορητής υπολογιστικής ικανότητας παραμένει ανοιχτή στέλνοντας δεδομένα από το επιταχυνσιόμετρο κάθε 200 millisecond καθ' όλη τη διάρκεια της χρήσης. Ο χρήστης φέρνει το χέρι του στο ύψος του στήθους που είναι η αρχική θέση για την εκτέλεση όλων των χειρονομιών. Με το πάτημα του κουμπιού στη διεπαφή της εφαρμογής της συσκευής κινητού υπολογισμού ξεκινά η καταγραφή των δεδομένων στη συσκευή μέσα στο χρονικό παράθυρο των 5 δευτερολέπτων που έχει οριστεί, στο οποίο χρονικό παράθυρο ο χρήστης εκτελεί τη χειρονομία και μένει στην τελική θέση μέχρι να εμφανιστεί στην οθόνη το μήνυμα επιβεβαίωσης της επιτυχούς καταγραφής.

Από τη στιγμή που η χειρονομία έχει ολοκληρωθεί ο χρήστης έχει στη διάθεσή του πέντε επιλογές:

- Αναγνώριση της χειρονομίας με τον αλγόριθμο FFT.
- Αναγνώριση της χειρονομίας με τον αλγόριθμο SAX με το global data set.
- Αναγνώριση της χειρονομίας με τον αλγόριθμο SAX με το προσωποκεντρικό data set.
- Αποθήκευση της χειρονομίας σε αρχείο για την εκπαίδευση του προσωποκεντρικού data set του αλγορίθμου SAX.
- Διαγραφή της τρέχουσας χειρονομίας και καταγραφή νέας.

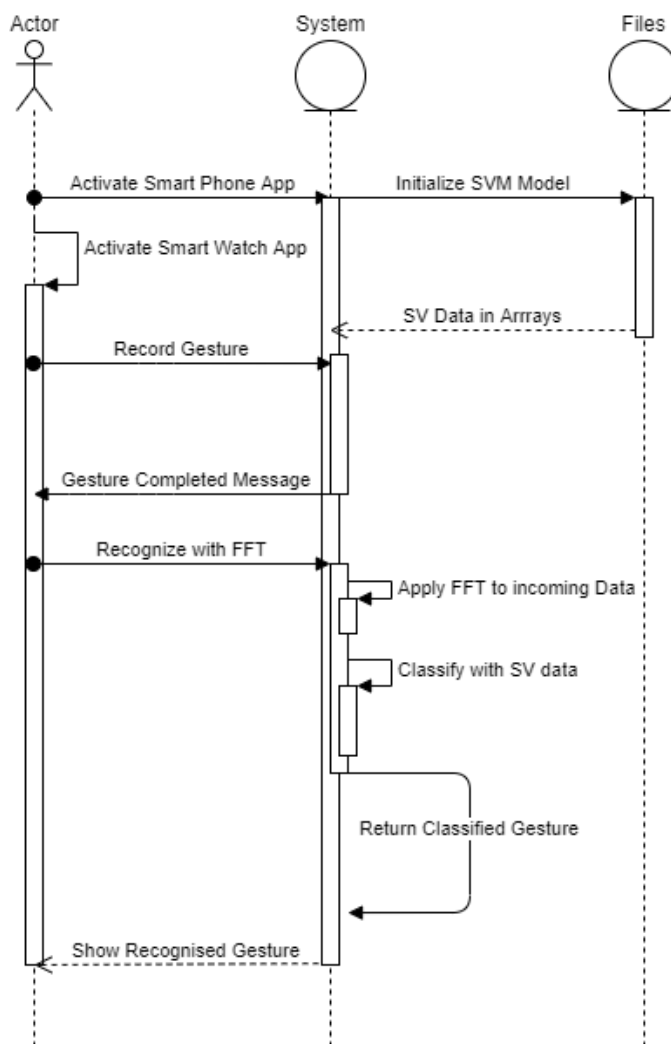


ΔΙΑΓΡΑΜΜΑ 4.1.1: ΔΡΔ που δείχνει τον τρόπο χρήσης του συστήματος.

4.2 Λογισμικό

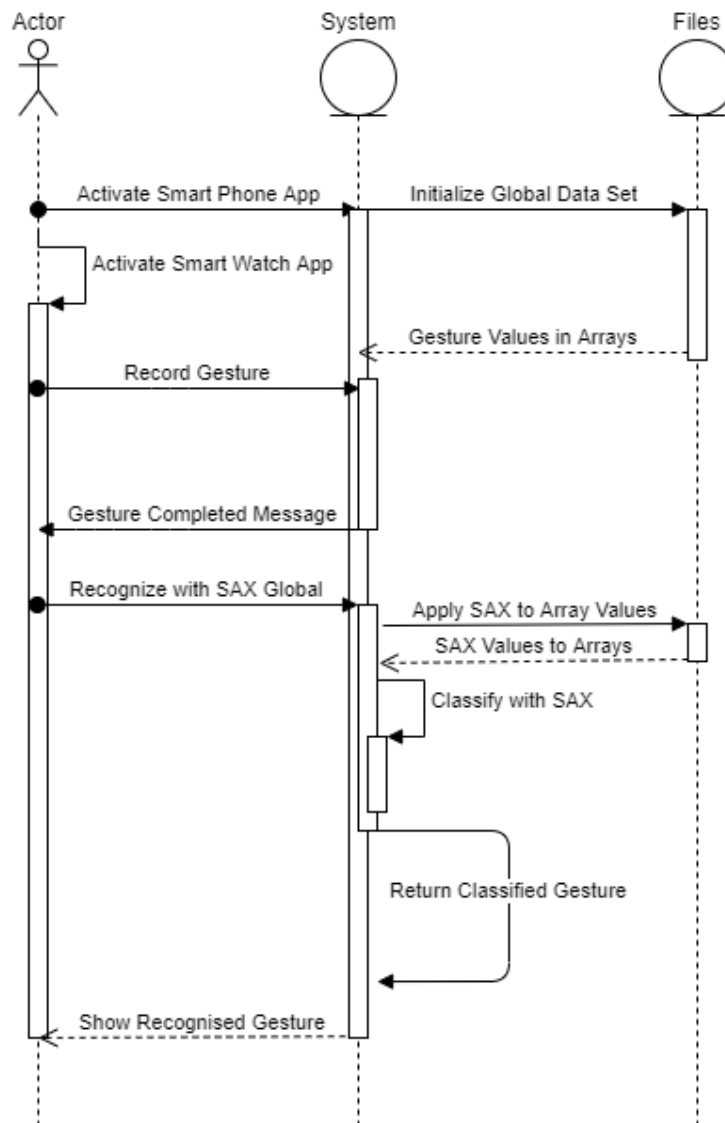
Σύμφωνα με τις πέντε επιλογές που έχει ο χρήστης στη διάθεσή του, όπως αναφέρθηκαν παραπάνω, το σύστημα χωρίζεται στις εξής λειτουργίες που παρουσιάζονται με τα διαγράμματα ακολουθίας [19] τους που δείχνουν πως επικοινωνούν οι οντότητες του συστήματος και τα δεδομένα που ανταλλάσσουν. Στον οριζόντιο άξονα απεικονίζονται οι οντότητες του συστήματος και στον κατακόρυφο απεικονίζεται ο χρόνος.

- **ΔΙΑΓΡΑΜΜΑ 4.2.1: Λειτουργία αναγνώρισης της χειρονομίας με τον αλγόριθμο FFT με το SVM μοντέλο που δημιουργήθηκε με το global data set.**



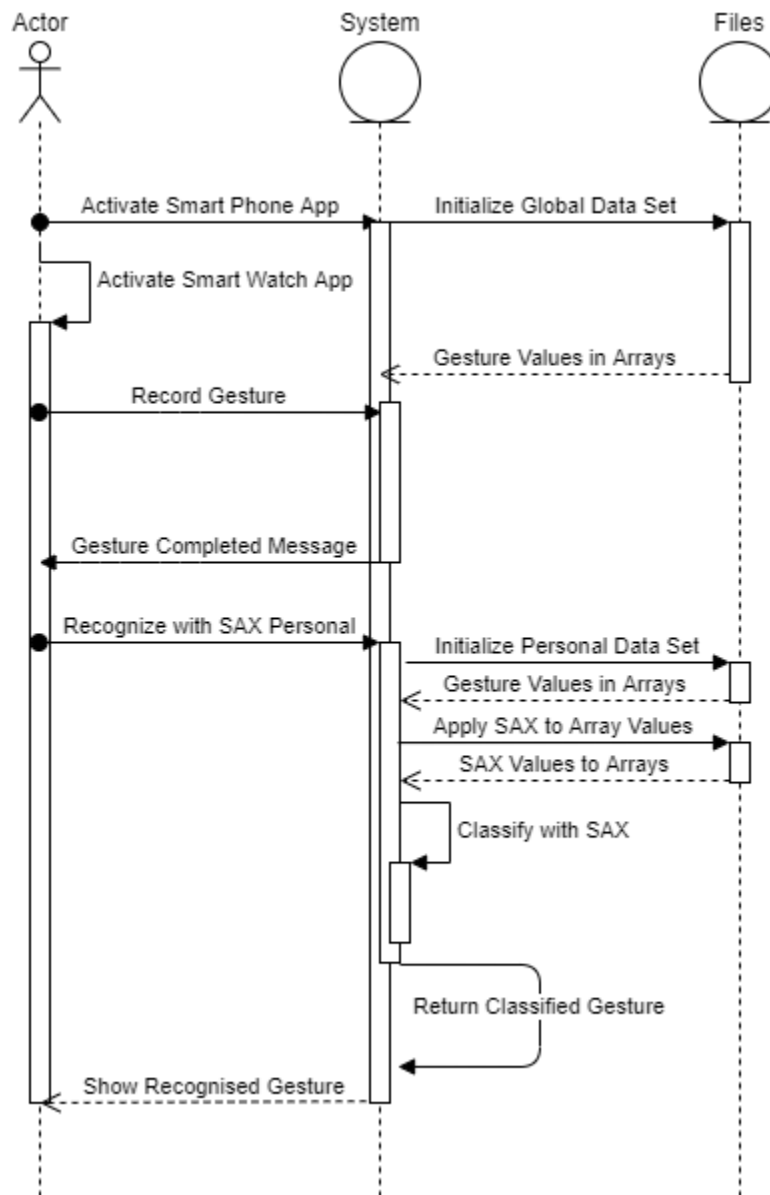
Στη λειτουργία αναγνώρισης με τον αλγόριθμο FFT ο χρήστης εφόσον έχει ενεργοποιήσει και τις 2 εφαρμογές, στη συσκευή έχει γίνει αρχικοποίηση των δεδομένων του μοντέλου του SVM από το αρχείο. Με το πάτημα της καταγραφής της χειρονομίας τα δεδομένα καταγράφονται από την εφαρμογή και όταν επιλεγθεί η αναγνώριση με FFT ο αλγόριθμος κατηγοριοποιεί τη χειρονομία και εμφανίζει το αποτέλεσμα στην οθόνη της διεπαφής της εφαρμογής.

- **ΔΙΑΓΡΑΜΜΑ 4.2.2: Λειτουργία αναγνώρισης της χειρονομίας με τον αλγόριθμο SAX με το global data set.**



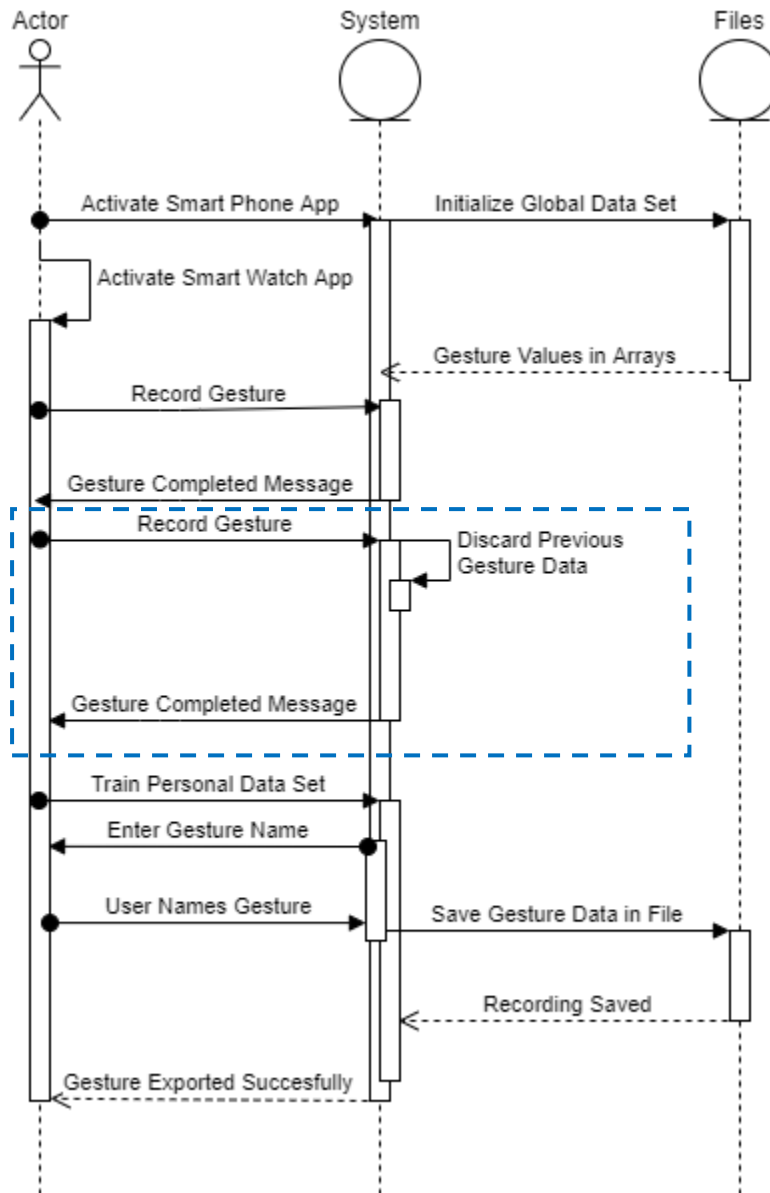
Στη λειτουργία αναγνώρισης με τον αλγόριθμο SAX με το global data set το μόνο που διαφέρει από την προηγούμενη λειτουργία είναι το σημείο που η εφαρμογή της συσκευής κινητού υπολογισμού μετασχηματίζει με τον αλγόριθμο SAX τα «ωμά» δεδομένα που εισέρχονται με τη νέα χειρονομία και αυτά που περιέχουν οι πίνακες της μετά την αρχικοποίηση τους στην έναρξη της και τα αποθηκεύει σε ένα αρχείο. Από αυτό το αρχείο τα διαβάζει κάθε φορά που θέλει να συγκρίνει την κάθε νέα καταγεγραμμένη χειρονομία από το χρήστη με τις αποθηκευμένες του data set και μετά την κατηγοριοποίησή της εμφανίζει το αποτέλεσμα στη διεπαφή της εφαρμογής.

- **ΔΙΑΓΡΑΜΜΑ 4.2.3: Λειτουργία αναγνώρισης της χειρονομίας με τον αλγόριθμο SAX με το προσωποκεντρικό data set.**



Στη λειτουργία αναγνώρισης με τον αλγόριθμο SAX με το προσωποκεντρικό data set που μπορεί να εκπαιδεύει ο χρήστης, η διαφορά με τη λειτουργία που περιεγράφηκε προηγουμένως είναι η αρχικοποίηση των τιμών που περιέχει το προσωποκεντρικό data set σε πίνακες καθώς στην έναρξη της εφαρμογής αρχικοποιούνται μόνο οι τιμές του global data set. Στη συνέχεια περνούν από μετασχηματισμό SAX και αυτές και αποθηκεύονται σε ένα νέο αρχείο για γρήγορη προσπέλαση. Η αρχικοποίηση συμβαίνει κάθε φορά που έχει αποθηκευτεί νέα χειρονομία στο αρχείο με το προσωποκεντρικό data set μετά την εκπαίδευση. Από το αρχείο αυτό διαβάζει κάθε φορά τα δεδομένα για να κατηγοριοποιήσει τη νέα εισερχόμενη χειρονομία του χρήστη.

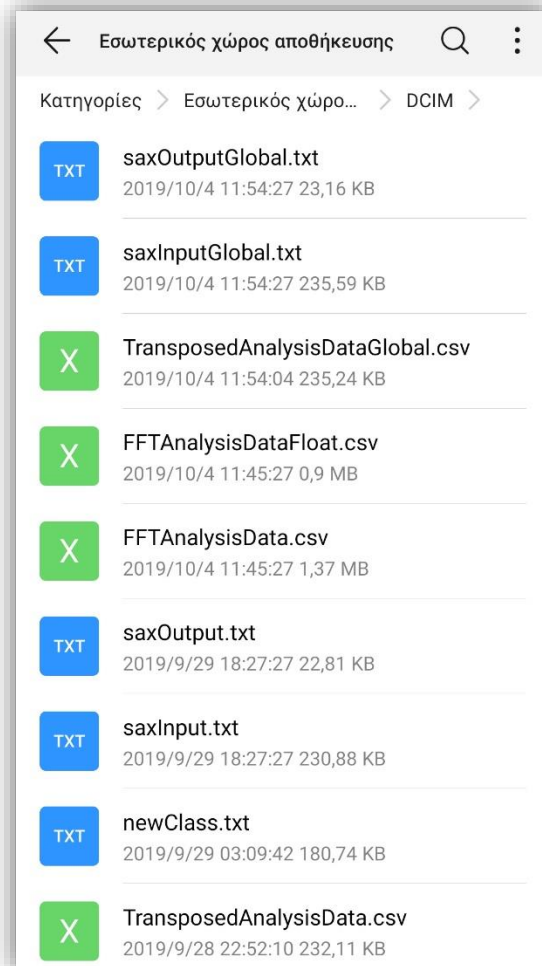
- **ΔΙΑΓΡΑΜΜΑ 4.2.4: Λειτουργία καταγραφής της χειρονομίας και εκπαίδευσης του προσωποκεντρικού data set.**



Με τη λειτουργία καταγραφής της χειρονομίας ο χρήστης δίνει εντολή στη συσκευή κινητού υπολογισμού να αρχίσει να αποθηκεύει προσωρινά τα δεδομένα, μέχρι την ολοκλήρωση της χειρονομίας, σε έναν πίνακα. Του δίνεται επίσης η επιλογή αν δεν είναι ικανοποιημένος με την εκτέλεση της χειρονομίας να διαγραφούν τα δεδομένα της όπως φαίνεται στο παραλληλόγραμμο με μπλε διακεκομμένες γραμμές και να γίνει νέα καταγραφή με το πάτημα του ίδιου κουμπιού. Στη συνέχεια εφόσον πατήσει το κουμπί για εκπαίδευση η εφαρμογή τον προτρέπει να εισάγει το όνομα της χειρονομίας που εκτέλεσε και αφού το εισάγει η χειρονομία αποθηκεύεται στο αρχείο με το προσωποκεντρικό data set και εμφανίζεται μήνυμα επιτυχίας.

4.3 Δομή Αρχείων Αποθήκευσης

Στην υποενότητα αυτή θα εξηγηθεί ο τρόπος αποθήκευσης των δεδομένων των χειρονομιών και τα αρχεία που δημιουργεί το σύστημα και χρειάζεται για τη λειτουργία του όπως εμφανίζονται στην εικόνα που ακολουθεί.



ΕΙΚΟΝΑ 4.3.1: Τα αρχεία που χρησιμοποιεί η εφαρμογή

Για την αποθήκευση των αρχείων επιλέχθηκε ο εσωτερικός χώρος αποθήκευσης της συσκευής κινητού υπολογισμού και συγκεκριμένα ο φάκελος DCIM. Στην πρώτη εγκατάσταση της εφαρμογής στη συσκευή ζητείται άδεια από το χρήστη για την πρόσβαση στον εσωτερικό χώρο αποθήκευσης.

Η κάθε εγγραφή χειρονομίας αποτελείται από 90 τιμές ($30x + 30y + 30z$) με αυτή τη σειρά και στο τέλος της έχει ένα πεδίο με το όνομα της κάθε χειρονομίας.

Κατά τη δημιουργία του data set που αποτελείται από τις 250 χειρονομίες 5 ατόμων δημιουργήθηκε το αρχείο **TransposedAnalysisData.csv** με τη βοήθεια της λειτουργίας εκπαίδευσης που αναλύθηκε προηγουμένως και οι χειρονομίες ταξινομήθηκαν χειροκίνητα στο Open Office Calc με την εξής σειρά: hh, hu, hud, hh2, hu2.

Βάσει αυτού του αρχείου δημιουργήθηκε το αρχείο **TransposedAnalysisDataGlobal.csv** που περιέχει τις 250 χειρονομίες που προαναφέρθηκαν και μπορεί ο κάθε νέος χρήστης να προσθέσει τις χειρονομίες που εκτελεί σε αυτό, είτε μπορεί να το διαγράψει και την επόμενη φορά που θα χρησιμοποιήσει τη λειτουργία εκπαίδευσης της εφαρμογής θα δημιουργηθεί ένα νέο που θα περιέχει μόνο όσες προσωπικές χειρονομίες έχει αποφασίσει να αποθηκεύσει. Σε αυτό το αρχείο οι χειρονομίες δεν είναι ταξινομημένες καθώς ο χρήστης τις εκτελεί και τις αποθηκεύει με όποια σειρά επιθυμεί.

Η μορφή των δεδομένων του επιταχυνσιόμετρου μέσα στα δύο παραπάνω αρχεία .csv είναι η εξής (εμφανίζονται ενδεικτικά οι πρώτες 3 στήλες των συνιστωσών και η στήλη της κλάσης της χειρονομίας).

X1	X2	X3	Y1	Y2	Y3	Z1	Z2	Z3	Gesture
-0.3589234	-0.35174492	-0.39481574	-2.5292134	-2.3425734	-1.6319051	9.600004	9.595219	9.492328	hh
-1.1078769	-1.2131611	-1.1365907	-2.428715	-2.4717858	-2.3688943	9.499506	9.643075	9.420543	hh
-1.3256238	-1.2394822	-1.3136597	-4.1635113	-4.1850467	-4.1587257	8.925228	8.889337	9.099904	hu
-1.146162	-1.2251252	-1.2083755	3.9218364	3.7686956	3.9673	8.781659	8.874979	8.817552	hu
-0.37806597	-0.32063824	-0.33020952	-0.7178468	-0.8111669	-0.6963114	9.810573	9.79143	9.810573	hud
-1.0743774	-1.1772687	-1.2586247	-1.9716859	-1.7515461	-1.1581261	9.65504	9.616755	9.482756	hud
-0.6867401	-0.8877372	-2.2325034	-1.4045869	-0.46660042	1.1294123	9.652647	9.690931	8.91805	hh2
-0.55034924	-0.60777694	-0.6125626	-5.6781683	-5.575277	-5.5704913	8.190632	8.114061	8.142776	hh2
-0.87338024	-0.9236295	-0.849452	4.328616	4.3669014	4.240082	8.647661	8.673982	8.693125	hu2
-0.66759753	-0.6221339	-0.69152576	-0.9236295	-1.1318051	-1.0241281	9.772287	9.79143	9.803394	hu2

ΕΙΚΟΝΑ 4.3.2: Η μορφή των δεδομένων μες το αρχείο *TransposedAnalysisData.csv*

Τα αρχεία **FFTAnalysisData.csv** και **FFTAnalysisDataFloat.csv** δημιουργήθηκαν κατά τη διάρκεια της ανάπτυξης της εφαρμογής για δύο λόγους. Πρώτος λόγος ήταν ο έλεγχος του επιτυχούς μετασχηματισμού τους και του υπολογισμού του μέτρου τους από τον αλγόριθμο FFT και ο δεύτερος λόγος ήταν η εισαγωγή των δεδομένων στο RStudio για τη δημιουργία γραφικών παραστάσεων που απεικονίζουν την κατανομή του και για την εκπαίδευση του SVM.

Παρατίθενται δύο εικόνες με ενδεικτικές τιμές των περιεχομένων του κάθε αρχείου.

1	151.001464 87.52784957953574 + 168.11640957050093i	-73.14970231229927 + 78.74251892904775i	hh
2	154.1288835 81.94974884250303 + 165.9712871545461i	-75.99233228091865 + 83.28116276061692i	hh

ΕΙΚΟΝΑ 4.3.3: Η μορφή των δεδομένων μες το αρχείο FFTAnalysisData.csv

X1	X2	X3	Y1	Y2	Y3	Z1	Z2	Z3	Gesture
3.2247772	2.7407966	1.4572058	0.18942007	0.14098343	0.19096687	0.14098343	0.18942007	0.19668807	hu
3.3993037	2.6191552	1.4240795	0.11139858	0.16326047	0.07511284	0.16326047	0.11139858	0.07126559	hu

ΕΙΚΟΝΑ 4.3.4: Η μορφή των δεδομένων μες το αρχείο FFTAnalysisDataFloat.csv

Το αρχείο **newClass.txt** περιέχει τις αποθηκευμένες μεταβλητές του μοντέλου του SVM μαζί με τους Support Vectors όπως υπολογίστηκαν από την εκπαίδευση του συστήματος με την εισαγωγή των παραπάνω δεδομένων στο RStudio. Χρησιμοποιείται για την αρχικοποίηση του SVM της εφαρμογής κατά την έναρξή της.

```

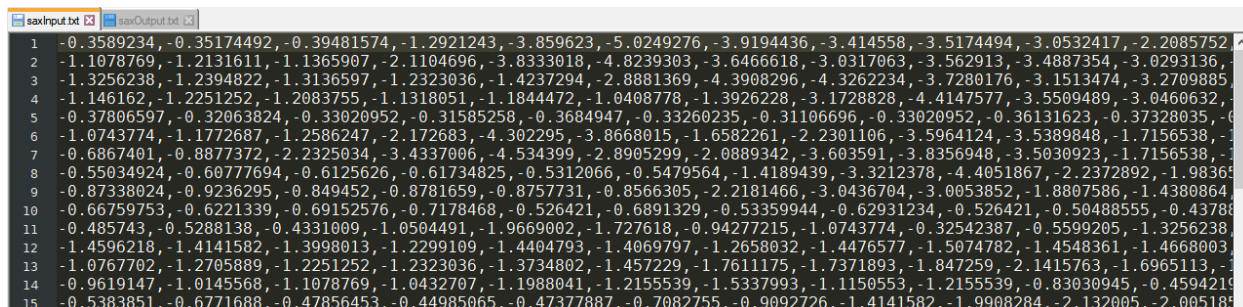
1 svm_type c_svc
2 kernel_type rbf
3 gamma 0.02
4 nr_class 5
5 total_sv 193
6 rho 0.8978307077324722 0.82670111292684634 2.0873273180864893 0.85280714751238074 0.23075687600590519 0.28368425072817116
7 label 1 3 5 2 4
8 nr_sv 24 38 41 44 46
9 SV
10 0 6.2835501439784389 0 1:0.075496613 2:0.78876614 3:0.67246645 4:-0.49030632 5:0.13317688 6:-0.59154633 7:0.04110317 8:
11 0.25452844470684627 0.92514818345247618 0 0.70132431738774914 1:0.11979945 2:0.67431813 3:0.83321685 4:-0.54481467 5:0.30
12 0 0.12656201265461012 0 1:-0.39834308 2:0.96952773 3:0.87404373 4:-0.52305706 5:0.59003718 6:-1.1668939 7:-0.62477098 8:
13 0 2.5212572425872359 0 1:1.4173924 2:-0.14512769 3:0.03157805 4:-0.50966903 5:-0.89378856 6:-0.693851 7:0.68540441 8:
14 0.50729100196064904 0.2031355603155735 5.4213173742448273 0 1:0.30314551 2:0.95258623 3:0.087654713 4:-0.39385427 5:-0.79
15 0.062347048792510608 0 0.89504200045999779 0 1:0.7828494 2:0.58601791 3:-0.052660059 4:-0.6705354 5:-0.72145963 6:-0.2810
16 0.47868005185394033 0.29828514822593316 2.0272191463327762 0.47984076214445986 1:1.6592445 2:-0.28996391 3:-0.83618232 4:
17 0 0.70496370255702212 0.0045474624098850915 1:0.63417924 2:0.80630681 3:-0.067293308 4:-0.25360629 5:-0.96518156 6:-0.0
18 1.6478653813569137 0.80717913857019452 0 0.94229791347417435 1:-1.491574 2:1.6333335 3:0.40325893 4:-0.5879685 5:-0.13652
19 0.89035472880374955 0.90857374524494094 2.2342961307009892 0.93634222422758295 1:1.3473282 2:0.077183918 3:-0.44322531 4:
20 1.0901638465814516 0 1.8379821860116727 0 1:0.71658155 2:0.66461751 3:-0.013286162 4:-0.74152789 5:-0.40369179 6:-0.32318
21 0 0.50301148083724256 0 1:1.5766044 2:-0.53702322 3:0.0054011859 4:-0.77082621 5:-0.80349045 6:-0.78242725 7:-1.1308352
22 0.94971630064540191 1.0748313145237351 1.6439445792378342 0.8289427602158248 1:0.67275333 2:0.57170209 3:0.42052788 4:-0.
23 0 5.5554893502931755 0 1:-0.68442982 2:1.4124933 3:0.38325149 4:-0.61364026 5:-0.037483815 6:-0.48201792 7:-0.25576076
24 1.6217901260625993 1.277107007333369 2.9562591013135981 0.75597662977446822 1:0.1097659 2:0.9728064 3:0.38425949 4:-0.185
25 0 0.39033611821780551 0 1:1.4995237 2:-0.25767071 3:-0.18985174 4:-0.55850319 5:-0.98354185 6:-0.3443409 7:-0.50884671
26 0.51188985049334979 0.35797272566051108 0 0.12157687341028761 1:0.49035613 2:0.63315165 3:0.43202573 4:-0.38315717 5:-0.4
27 0 1.2663932725748637 0 1:0.49594906 2:0.87683785 3:0.16380139 4:-0.50223721 5:-0.4459763 6:-0.62085101 7:-0.57273842 8:
28 0.15599636111606976 3.0446086219691764 0.0024763749910670795 1:-1.6183132 2:1.5253328 3:0.62421712 4:-0.517112 5:-0.076
29 0 4.3177146555760819 0.059778822797099875 1:1.3924785 2:0.0081233151 3:-0.39282709 4:-0.56610429 5:-1.0423313 6:-0.6684
30 0 1.6288503881841687 0 1:0.49201706 2:0.8314828 3:0.25002088 4:-0.35997406 5:-0.53712744 6:-0.7158452 7:-0.34227219 8:
31 0.96591753649476697 0 4.1469719989972136 0 1:0.35544808 2:0.82460895 3:0.22277723 4:-0.68837227 5:-0.25576367 6:-0.259093
32 0.42199219769205287 0.15318121316887476 2.4985124216529409 0 1:-1.8659951 2:1.2518563 3:0.88132977 4:-0.64644266 5:0.0763
33 0 0.01532852367096318 0 0 1:1.6879544 2:-0.59153091 3:-0.23867992 4:-0.62073589 5:-1.0444597 6:-0.32643807 7:-0.89482993
34 -0 0 0.53048242060138262 1:0.15603474 2:0.63048982 3:-1.2492257 4:-0.30617737 5:-0.78559049 6:0.61489759 7:0.1319289 8:
35 -0 0.87754148457279169 0 1:-0.4300703 2:0.62108358 3:-1.638789 4:0.043947939 5:0.85422306 6:2.052496 7:0.59061581 8:1.0
36 -0.079235830886907901 1.7939181298981648 0.49415560552845822 1.2633447366709269 1:0.58296153 2:-0.48335873 3:0.11514871 4:
37 -0 0.0070151437972472532 0 0 1:0.27613009 2:0.56471218 3:0.85310543 4:-0.55056425 5:-1.0018256 6:0.14649274 7:-0.2779653
38 -0 0.98079148725960175 0 3.5450043587868243 1:0.32483945 2:0.41091005 3:-0.84896955 4:-0.69627488 5:-1.0673205 6:-0.20529
39 -0 0 1.1361778301839316 1:1.7251391 2:-1.1543677 3:-1.3642685 4:-0.79111985 5:-0.20609415 6:0.27428254 7:-0.26106605 8:
40 -0 0 0.095790598664945437 1:0.64567009 2:-0.83872131 3:-1.0764106 4:-0.8951762 5:2.5248879 6:2.3373928 7:-0.20554575 8:
41 -0.1246269536019433 1.6885784896828395 1.1138230034165495 1.0650478962177923 1:-2.0467314 2:-0.061629081 3:0.17176732 4:
42 -0 0.97181639530182606 0 0.48864216526854648 1:1.4418996 2:-2.7250675 3:-1.8913511 4:-0.45465896 5:2.9592582 6:2.3385685
43 -0.20279807951335802 0 0 1:1.0537165 2:-1.3007655 3:-0.24152062 4:-0.7303667 5:2.351894 6:0.8838637 7:-1.2951439 8:-0.146
44 -0 0.83596394807595509 0 2.559733053055529 1:1.1016123 2:-0.19641003 3:-0.49107148 4:-0.71552888 5:-1.4326213 6:-1.54251

```

ΕΙΚΟΝΑ 4.3.5: Η μορφή των δεδομένων μες το αρχείο newClass.txt

Τα αρχεία **saxInput.txt** και **saxInputGlobal.txt** περιέχουν τα «ωμά» δεδομένα από το επιταχυνσιόμετρο όπως και τα αρχεία **TransposedAnalysisData.csv** και **TransposedAnalysisDataGlobal.csv** αντίστοιχα με τη διαφορά του ότι λείπει η πρώτη εγγραφή που περιγράφει ποια τιμή του επιταχυνσιόμετρου βρίσκεται στην κάθε στήλη όπως λείπει επίσης και η τελευταία στήλη που περιγράφει την κλάση που ανήκει η κάθε εγγραφή. Αυτό συμβαίνει καθώς η βιβλιοθήκη που χρησιμοποιήθηκε, για το μετασχηματισμό των χρονοσειρών σε αναπαράσταση SAX, λειτουργούσε μόνο με αρχεία .txt αυτής της μορφής. Η βιβλιοθήκη αναλαμβάνει την προσπέλαση των δεδομένων από αυτά τα αρχεία και αφού τα μετασχηματίσει αποθηκεύει τις αναπαραστάσεις SAX στα αρχεία **saxOutput.txt** και **saxOutputGlobal.txt** αντίστοιχα από τα οποία γίνεται στη συνέχεια η αρχικοποίηση των πινάκων της εφαρμογής που αναλαμβάνουν τη σύγκριση των δεδομένων της εισερχόμενης χειρονομίας με τα δεδομένα των αρχείων που αρχικοποιήθηκαν.

Παρατίθενται εικόνες με ενδεικτικές τιμές από τα αρχεία **saxInput.txt** και **saxOutput.txt** καθώς το πλήθος των τιμών δεν επιτρέπει την εμφάνιση όλων των τιμών που περιέχουν όπως και στις προηγούμενες περιπτώσεις και τα δεδομένα που περιέχουν είναι στην ίδια μορφή με τα δεδομένα των αρχείων **saxInputGlobal.txt** και **saxOutputGlobal.txt**.

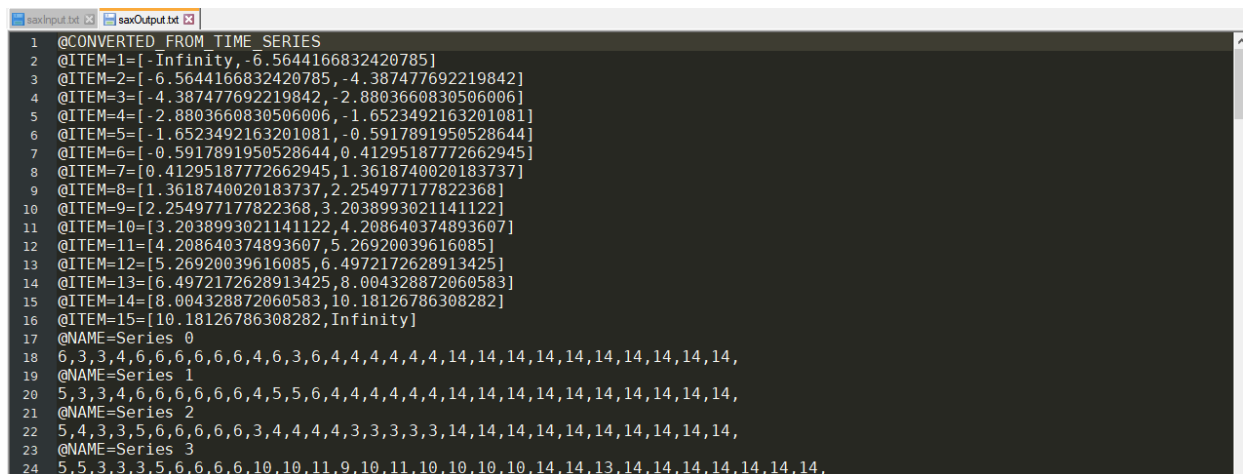


```

1 -0.3589234,-0.35174492,-0.39481574,-1.2921243,-3.859623,-5.0249276,-3.9194436,-3.414558,-3.5174494,-3.0532417,-2.2085752
2 -1.1078769,-1.2131611,-1.1365907,-2.1104696,-3.8333018,-4.8239303,-3.6466618,-3.0317063,-3.562913,-3.4887354,-3.0293136
3 -1.3256238,-1.2394822,-1.3136597,-1.2323036,-1.4237294,-2.8881369,-4.3908296,-4.3262234,-3.7280176,-3.1513474,-3.2709885
4 -1.146162,-1.2251252,-1.2083755,-1.1318051,-1.1844472,-1.0408778,-1.3926228,-3.1728828,-4.4147577,-3.5509489,-3.0460632
5 -0.37806597,-0.32063824,-0.33020952,-0.31585258,-0.3684947,-0.33260235,-0.31106696,-0.33020952,-0.36131623,-0.37328035,-0.37328035
6 -1.0743774,-1.1772687,-1.2586247,-2.172683,-4.302295,-3.8668015,-1.6582261,-2.2301106,-3.5964124,-3.5389848,-1.7156538
7 -0.6867401,-0.8877372,-2.2325034,-3.4337006,-4.534399,-2.8905299,-2.0889342,-3.603591,-3.8356948,-3.5030923,-1.7156538
8 -0.55034924,-0.60777694,-0.6125626,-0.61734825,-0.5312066,-0.5479564,-1.4189439,-3.3212378,-4.4051867,-2.2372892,-1.98365
9 -0.87338024,-0.9236295,-0.849452,-0.8781659,-0.8757731,-0.8566305,-2.2181466,-3.0436704,-3.0053852,-1.8807586,-1.4380864
10 -0.66759753,-0.6221339,-0.69152576,-0.7178468,-0.526421,-0.6891329,-0.53359944,-0.62931234,-0.526421,-0.50488555,-0.43780
11 -0.485743,-0.5288138,-0.4331009,-1.0504491,-1.9669002,-1.727618,-0.94277215,-1.0743774,-0.32542387,-0.5599205,-1.3256238
12 -1.4596218,-1.4141582,-1.3998013,-1.2299109,-1.4404793,-1.4069797,-1.2658032,-1.4476577,-1.5074782,-1.4548361,-1.4668003
13 -1.0767702,-1.2705889,-1.2251252,-1.2323036,-1.3734802,-1.457229,-1.7611175,-1.7371893,-1.847259,-2.1415763,-1.6965113
14 -0.9619147,-1.0145568,-1.1078769,-1.0432707,-1.1988041,-1.2155539,-1.5337993,-1.1150553,-1.2155539,-0.83030945,-0.4594219
15 -0.5383851,-0.6771688,-0.47856453,-0.44985065,-0.47377887,-0.7082755,-0.9092726,-1.4141582,-1.9908284,-2.132005,-2.00518

```

ΕΙΚΟΝΑ 4.3.6: Η μορφή των δεδομένων μες το αρχείο **saxInput.txt**



```

1 @CONVERTED FROM TIME SERIES
2 @ITEM=1=[-Infinity,-6.5644166832420785]
3 @ITEM=2=[-6.5644166832420785,-4.387477692219842]
4 @ITEM=3=[-4.387477692219842,-2.8803660830506006]
5 @ITEM=4=[-2.8803660830506006,-1.6523492163201081]
6 @ITEM=5=[-1.6523492163201081,-0.5917891950528644]
7 @ITEM=6=[-0.5917891950528644,0.41295187772662945]
8 @ITEM=7=[0.41295187772662945,1.3618740020183737]
9 @ITEM=8=[1.3618740020183737,2.254977177822368]
10 @ITEM=9=[2.254977177822368,3.2038993021141122]
11 @ITEM=10=[3.2038993021141122,4.208640374893607]
12 @ITEM=11=[4.208640374893607,5.26920039616085]
13 @ITEM=12=[5.26920039616085,6.4972172628913425]
14 @ITEM=13=[6.4972172628913425,8.004328872060583]
15 @ITEM=14=[8.004328872060583,10.18126786308282]
16 @ITEM=15=[10.18126786308282,Infinity]
17 @NAME=Series 0
18 6,3,3,4,6,6,6,6,6,6,4,6,3,6,4,4,4,4,14,14,14,14,14,14,14,14,14,14,
19 @NAME=Series 1
20 5,3,3,4,6,6,6,6,6,6,4,5,5,6,4,4,4,4,14,14,14,14,14,14,14,14,14,14,
21 @NAME=Series 2
22 5,4,3,3,5,6,6,6,6,6,3,4,4,4,3,3,3,3,14,14,14,14,14,14,14,14,14,14,
23 @NAME=Series 3
24 5,5,3,3,3,5,6,6,6,6,6,10,10,11,9,10,11,10,10,10,10,14,14,13,14,14,14,14,14,14,14,

```

ΕΙΚΟΝΑ 4.3.7: Η μορφή των δεδομένων μες το αρχείο **saxOutput.txt**

ΚΕΦΑΛΑΙΟ 5: ΠΡΟΕΤΟΙΜΑΣΙΑ ΚΑΙ ΧΡΗΣΗ ΕΡΓΑΛΕΙΩΝ

ΑΝΑΠΤΥΞΗΣ

Σε αυτό το κεφάλαιο αναλύεται η προετοιμασία που πρέπει να γίνει για να ξεκινήσει η ανάπτυξη του συστήματος. Θα δούμε αναλυτικά τις συσκευές, τα εργαλεία και το λογισμικό που χρειάζεται να είναι διαθέσιμα όπως επίσης, τις χειρονομίες και τον ορθό τρόπο εκτέλεσής τους.

5.1 Οι Διακριτές Χειρονομίες

Οι πέντε χειρονομίες που επιλέχθηκαν σύμφωνα με το σενάριο που περιεγράφηκε στο κεφάλαιο 1.2 Σκοπιμότητα και Οφέλη και ο προτεινόμενος τρόπος εκτέλεσής τους, όσον αφορά την τροχιά που θα πρέπει να ακολουθεί το χέρι του χρήστη, φαίνεται αναλυτικά στις εικόνες του Παραρτήματος Α. Όλες οι χειρονομίες πρέπει να εκτελούνται με το έξυπνο ρολόι φορεμένο στο αριστερό χέρι.

- **Χειρονομία hh**

Ο χρήστης φέρνει τον καρπό του χεριού του λίγο πιο κάτω από το ύψος του διαφράγματος και το κρατάει σε απόσταση 10 εκατοστών περίπου από το σώμα του με προσανατολισμό της οθόνης του έξυπνου ρολογιού προς τα πάνω. Αυτή είναι η Αρχική Θέση (ΑΘ) για όλες τις χειρονομίες που θα ακολουθήσουν. Κρατώντας τον ώμο του σταθερό και τον καρπό του να διατηρεί τον προσανατολισμό της ΑΘ, κινεί το χέρι του με κατεύθυνση αριστερά και μακριά από το σώμα του στη φυσική ελλειπτική τροχιά του αγκώνα μέχρι να έρθει ο αγκώνας σε πλήρη έκταση. Στη συνέχεια επιστρέφει το χέρι του με την αντίστροφη κίνηση στην αρχική θέση μέχρι για να ολοκληρωθεί η χειρονομία.

- **Χειρονομία hu**

Ο χρήστης φέρνει τον καρπό του στην ΑΘ. Έχοντας σταθερό τον ώμο και τον αγκώνα του κινεί το χέρι του με κατεύθυνση πάνω και αριστερά μέχρι να έρθει η παλάμη του στο ύψος των ματιών του. Για να ολοκληρωθεί η χειρονομία, επιστρέφει το χέρι του ακολουθώντας την ίδια τροχιά με κατεύθυνση προς την αρχική θέση.

- **Χειρονομία hud**

Ο χρήστης φέρνει τον καρπό του στην ΑΘ. Ακολουθείται η ίδια τροχιά με το πρώτο βήμα της χειρονομίας hu. Όταν η παλάμη του χρήστη έρθει στο ύψος των ματιών, ο χρήστης εκτείνει τον αγκώνα του κινώντας τον καρπό του μακριά από το σώμα του και προς τα κάτω μέχρι να έρθει σε πλήρη έκταση. Η συγκεκριμένη χειρονομία ολοκληρώνεται σε αυτή τη θέση.

- **Χειρονομία hh2**

Ο χρήστης φέρνει το χέρι του στην ΑΘ. Στη συνέχεια εκτελεί τη χειρονομία hh δύο φορές χωρίς να κάνει παύση για να ξεκινήσει τη δεύτερη εκτέλεση καθώς η τελική θέση της hh είναι ίδια με την ΑΘ της οπότε η χειρονομία hh2 γίνεται μια ενιαία κίνηση.

- **Χειρονομία hu2**

Ο χρήστης φέρνει το χέρι του στην ΑΘ. Στη συνέχεια εκτελεί τη χειρονομία hu δύο φορές χωρίς να κάνει παύση για να ξεκινήσει τη δεύτερη εκτέλεση καθώς η τελική θέση της hu είναι ίδια με την ΑΘ της οπότε η χειρονομία hu2 γίνεται μια ενιαία κίνηση.

Υπενθυμίζεται πως ένας από τους σκοπούς ανάπτυξης ενός τέτοιου συστήματος είναι εφαρμογή του σε μια εγκατάσταση βοηθητικής διαβίωσης όπως ένα έξυπνο σπίτι για άτομα με κινητικά προβλήματα και το σύστημα μπορεί να προσαρμοστεί με βάση τις ανάγκες τους. Συνεπώς οι περιγραφές που προηγήθηκαν είναι ένας προτεινόμενος τρόπος λειτουργίας του συστήματος που αναπτύχθηκε, σύμφωνα με το προσωποκεντρικό αρχείο εκπαίδευσης και ο αναγνώστης ενθαρρύνεται να εκτελεί τις χειρονομίες με φυσικό τρόπο για την αποφυγή τραυματισμών.

5.2 Οι συσκευές που Χρησιμοποιήθηκαν

Για την επιλογή των συσκευών για την ανάπτυξη του συστήματος προτιμήθηκε, μετά από έρευνα, στη θέση της συσκευής κινητού υπολογισμού μια συσκευή που χρησιμοποιεί το λειτουργικό σύστημα Android καθώς οι χρήστες του καταλαμβάνουν το 76.24% της αγοράς των λειτουργικών στα έξυπνα τηλέφωνα όπως αναφέρει η ιστοσελίδα gs.statcounter.com[20] για το Σεπτέμβριο του 2019. Συγκεκριμένα, η εφαρμογή εγκαταστάθηκε και αξιολογήθηκε στο προσωπικό μου Smart Phone Huawei Honor 9 Lite με τα παρακάτω χαρακτηριστικά[21].

- Android 8.0 Oreo
- API Level 28
- CPU Octa-core (4x2.36 GHz Cortex-A53 & 4x1.7 GHz Cortex-A53)
- RAM 3GB
- Bluetooth 4.2, LE

Για την επιλογή της συσκευής φορητής υπολογιστικής ικανότητας προτιμήθηκε μια συσκευή που χρησιμοποιεί λειτουργικό σύστημα Wear OS (πρώην Android Wear) καθώς η Google που το ανέπτυξε προσφέρει ένα εύχρηστο API σε JAVA για την αξιόπιστη ανταλλαγή δεδομένων μέσω Bluetooth ανάμεσα στο Smart Watch και στο Smart Phone που έχουν συζευχθεί. Η επιλογή έγινε κυρίως για την ευκολία στην ανάπτυξη της επικοινωνίας των συσκευών και της μεταφοράς των δεδομένων αλλά και λόγω της ανόδου που θα έχουν οι συσκευές με λειτουργικό Wear OS, Android και Android Wear, στην αγορά, σύμφωνα με την ιστοσελίδα [statista.com](https://www.statista.com). [22] Συγκεκριμένα, η συσκευή στην οποία εγκαταστάθηκε η εφαρμογή που αναπτύχθηκε για το σύστημα είναι το προσωπικό μου Smart Watch Huawei Watch 2 Sport με τα εξής χαρακτηριστικά[23].

- Wear OS 2.0 (Πρώην Android Wear)
- API Level 28
- CPU Quad-core 1.1 GHz Cortex A7
- RAM 768MB
- Bluetooth 4.1, LE
- 3 Axis Accelerometer Sensor

5.3 Εργαλεία Ανάπτυξης

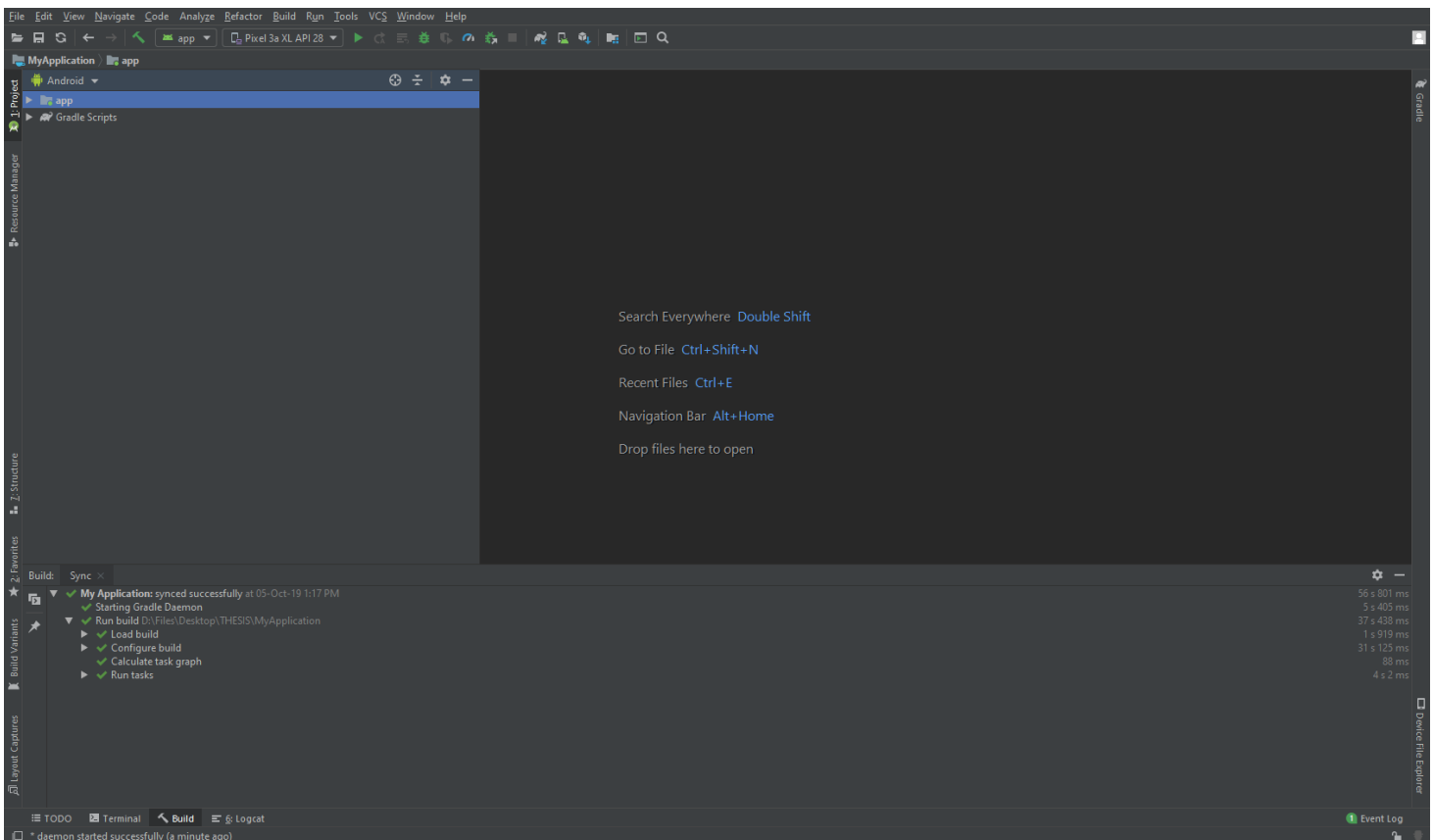
Στις παρακάτω υποενότητες αναφέρονται τα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη του συστήματος και ο τρόπος χρήσης τους.

5.3.1 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που δημιουργήθηκε από την Google για ανάπτυξη εφαρμογών Android που περιλαμβάνει τα πακέτα ανάπτυξης λογισμικού Android SDK και OpenJDK για την ανάπτυξη εφαρμογών σε Java. Το Android Studio είναι διαθέσιμο για λήψη σε λειτουργικά Windows, macOS και Linux στον παρακάτω σύνδεσμο:

<https://developer.android.com/studio/index.html>

Για τη δημιουργία μιας εφαρμογής Android ακολουθούνται τα βήματα που περιγράφονται στο Παράρτημα Β. Το Περιβάλλον εργασίας του Android Studio στο νέο Project φαίνεται στην εικόνα 5.3.1.1.



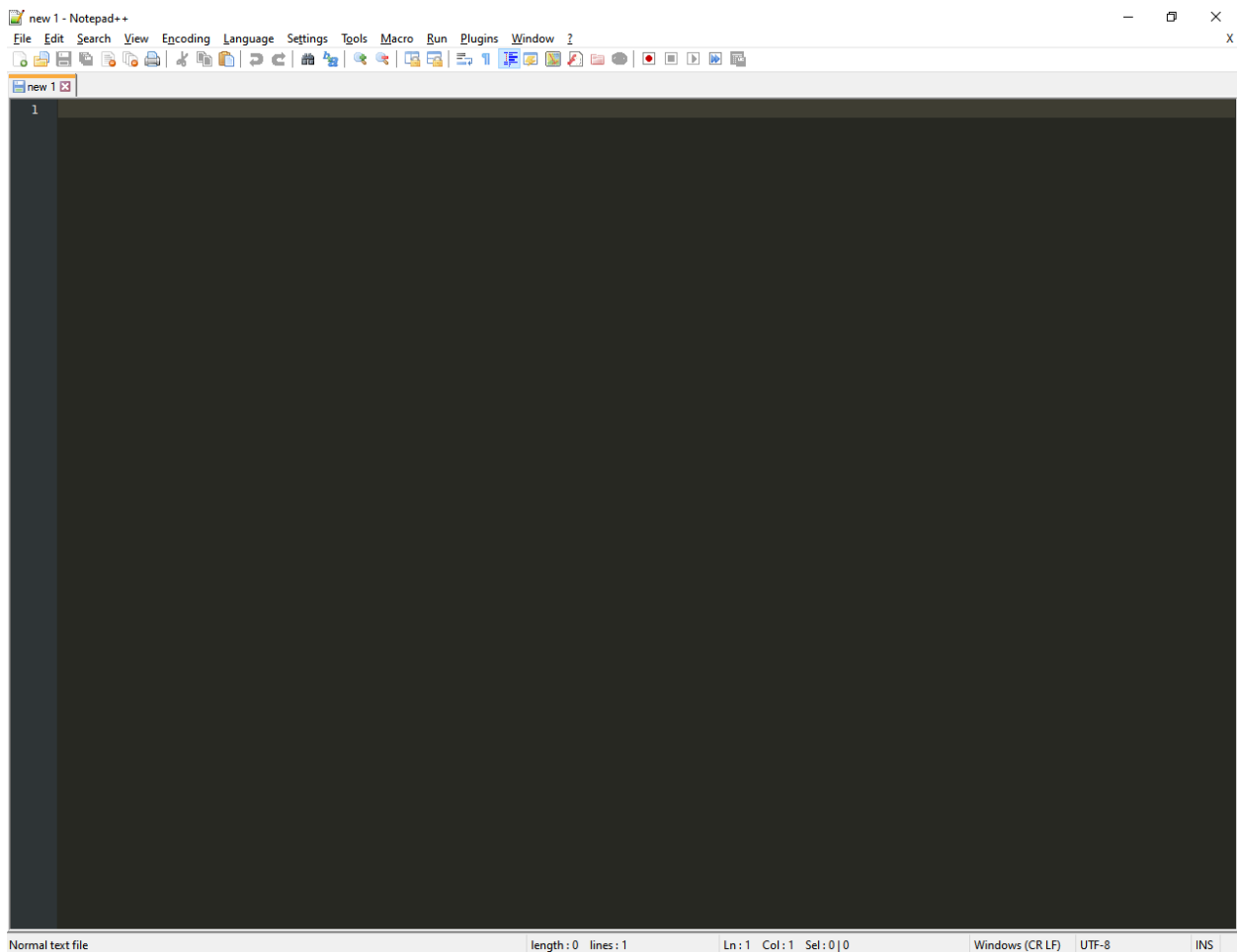
EIKONA 5.3.1.1: Android Studio

5.3.2 Notepad++

Το Notepad++ είναι μια παρόμοια εφαρμογή με το σημειωματάριο που προσφέρουν τα Windows που παρέχει περισσότερες λειτουργίες καθιστώντας την επεξεργασία των αρχείων πιο εύκολη και γρήγορη και χρησιμοποιήθηκε για τη μελέτη των δεδομένων που εξάχθηκαν σε αρχεία από την εφαρμογή Android. Το περιβάλλον εργασίας του φαίνεται στην παρακάτω εικόνα.

Το Notepad++ είναι διαθέσιμο για λήψη στον παρακάτω σύνδεσμο:

<https://notepad-plus-plus.org/downloads/>



EIKONA 5.3.2.1: Notepad++

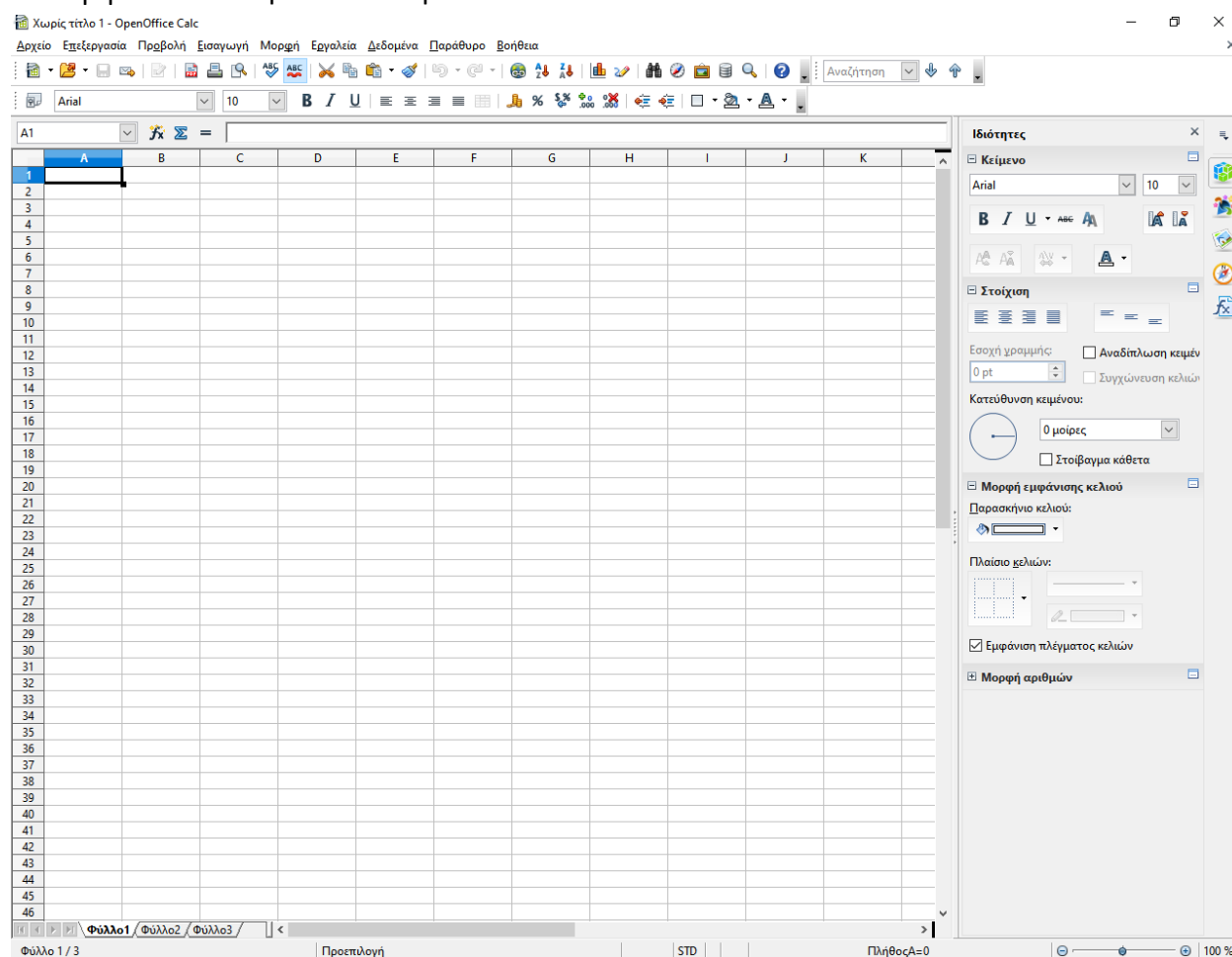
5.3.3 OpenOffice Calc

Το Apache OpenOffice είναι ένα ολοκληρωμένο πακέτο εφαρμογών γραφείου ανοιχτού κώδικα. Περιλαμβάνει: Επεξεργαστή Κειμένου (Writer), Υπολογιστικό Φύλλο (Calc), Δημιουργό Παρουσιάσεων (Impress), Πρόγραμμα Σχεδίασης (Draw), Βάση Δεδομένων (Base) και Επεξεργαστή Εξισώσεων (Math). Λειτουργεί σε όλες τις γνώστες πλατφόρμες λειτουργικών συστημάτων όπως Windows, Linux και Mac και είναι διαθέσιμο για λήψη ακολουθώντας το σύνδεσμο:

<https://www.openoffice.org/el/download/index.html>

Σε αυτή την εργασία χρησιμοποιήθηκε το Υπολογιστικό Φύλλο (Calc) για τη μελέτη των δεδομένων που εξάχθηκαν από την εφαρμογή Android σε αρχεία .csv.

Το περιβάλλον του φαίνεται στην εικόνα 5.3.3.1.



ΕΙΚΟΝΑ 5.3.3.1: Open Office Calc

5.3.4 RStudio

Το RStudio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη κώδικα με την, ανοιχτού κώδικα, γλώσσα προγραμματισμού R που χρησιμοποιείται κυρίως για υπολογιστική στατιστική και για γραφήματα και είναι διαθέσιμο για λήψη στις πλατφόρμες Windows, macOS και Linux στον εξής σύνδεσμο:

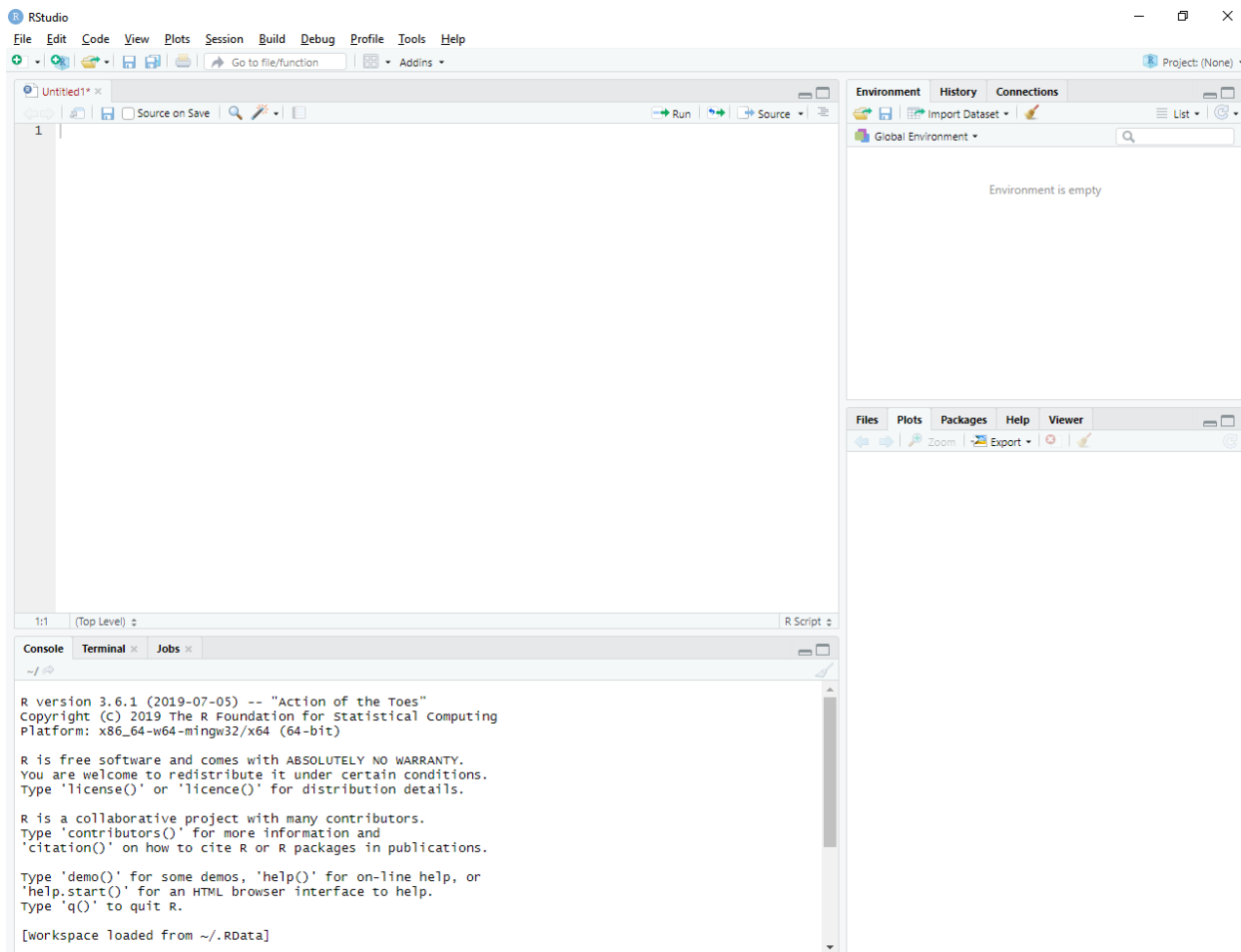
<https://rstudio.com/products/rstudio/download/>

Για τη χρήση του RStudio απαιτείται η λήψη της R από τον ακόλουθο σύνδεσμο:

<https://cran.r-project.org/mirrors.html>

Το RStudio χρησιμοποιήθηκε για τη δημιουργία γραφικών παραστάσεων και για την εκπαίδευση του SVM που χρησιμοποιεί ο αλγόριθμος FFT για την αναγνώριση των χειρονομιών.

Το περιβάλλον ανάπτυξης του RStudio φαίνεται στην εικόνα που ακολουθεί.



EIKONA 5.3.4.1: RStudio

5.4 Τεχνολογίες και Βιβλιοθήκες

5.4.1 Java



Η Java είναι μια αντικειμενοστραφής Γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρία Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε όλα τα λειτουργικά συστήματα χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.[24]

Η Java διαθέτει επίσης αυτόματη διαχείριση μνήμης χάρη στην ύπαρξη του συλλέκτη απορριμμάτων (Garbage Collector). Συλλογή απορριμμάτων είναι μία κοινή ονομασία που χρησιμοποιείται στον τομέα της πληροφορικής για να δηλώσει την ελευθέρωση τμημάτων μνήμης από δεδομένα που δε χρειάζονται και δε χρησιμοποιούνται άλλο. Αυτή η απελευθέρωση μνήμης στη Java είναι αυτόματη και γίνεται μέσω του συλλέκτη απορριμμάτων. Έτσι ο προγραμματιστής απαλλάσσεται από την διαδικασία απελευθέρωσης μνήμης, οπότε αποφεύγονται και σφάλματα δεικτών.

Όλα αυτά τα χαρακτηριστικά καθιστούν την Java μια πολύ καλή επιλογή για διάφορους τύπους λογισμικού και γι' αυτό προτιμήθηκε και στην ανάπτυξη αυτού του συστήματος καθώς υπάρχουν ήδη αρκετές βιβλιοθήκες με εργαλεία που μας είναι χρήσιμα, που μπορούν να ενσωματωθούν στο σύστημα χωρίς να εμφανιστούν προβλήματα φορητότητας.

5.4.2 XML



Η XML (eXtensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0, που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium).[25] Με την XML μπορούμε να διαμορφώσουμε τη διεπαφή (User Interface) στο διαδίκτυο και σε εφαρμογές Android. Για να αποκτήσει λειτουργικότητα το περιβάλλον της διεπαφής η XML συνδυάζεται με μια άλλη γλώσσα προγραμματισμού (στην προκειμένη με την Java) όπου το Android Studio επιτρέπει την ανάπτυξη κώδικα για την ενημέρωση των πεδίων της διεπαφής που περιγράφονται στο αρχείο .xml και τη γενικότερη αλληλεπίδραση του χρήστη με τη διεπαφή μέσω της Java.

5.4.3 SPMF



Η SPMF είναι μια βιβλιοθήκη ανοιχτού κώδικα για την εξόρυξη δεδομένων γραμμένη σε Java που αναπτύχθηκε από τον Καθηγητή Philippe Fournier-Viger και προσφέρει πάνω από 170 αλγόριθμους εξόρυξης δεδομένων.[26] Έγινε ενσωμάτωσή τμήματος της βιβλιοθήκης, μετά από κάποια επεξεργασία στον κώδικα της, στην εφαρμογή της συσκευής κινητού υπολογισμού καθώς περιείχε όλα τα απαραίτητα εργαλεία για το μετασχηματισμό των δεδομένων σε αναπαράσταση SAX, που είναι και ο δεύτερος αλγόριθμος αναγνώρισης χειρονομιών που επιλέχθηκε για το παρόν σύστημα.

Για τη λήψη της βιβλιοθήκης ακολουθείται ο σύνδεσμος:

<http://www.philippe-fournier-viger.com/spmf/index.php?link=download.php>

5.4.4 LibSvm

Η LibSvm είναι μια ολοκληρωμένη βιβλιοθήκη ανοιχτού κώδικα για κατηγοριοποίηση με μηχανές διανυσμάτων υποστήριξης (Support Vector Machines) από τους Chih-Chung και Chih-Jen Lin.[27]

Διατίθεται σε πολλές γλώσσες προγραμματισμού όπως Java, C++, Matlab και Python κ.ά. και είναι διαθέσιμη για λήψη στη διεύθυνση:

<https://www.csie.ntu.edu.tw/~cjlin/libsvm/> στην ενότητα “Download LIBSVM”

Η επιλογή για την ενσωμάτωσή της, στον κώδικα της εφαρμογής της συσκευής κινητού υπολογισμού, έγινε με βάση τον τρόπο κατηγοριοποίησης των χειρονομιών που προτιμήθηκε εφόσον παρέχει όλα τα εργαλεία που χρειάζονται για τη δημιουργία ενός μοντέλου SVM και την κατηγοριοποίηση των εισερχόμενων δεδομένων της νέας χειρονομίας. Χρησιμοποιήθηκε κατά τη λειτουργία αναγνώρισης με τον αλγόριθμο FFT καθώς σε παρόμοιες μελέτες η χρήση μιας SVM απέφερε μεγάλο ποσοστό ακρίβειας στην αναγνώριση χειρονομιών.

5.4.5 OpenCSV

Η OpenCSV είναι μια βιβλιοθήκη ανοιχτού κώδικα που παρέχει εύχρηστα εργαλεία διαχείρισης αρχείων CSV (Comma Separated Values), για Java. [28] Ενσωματώθηκε στον κώδικα της εργασίας από τη στιγμή που επιλέχθηκε η αποθήκευση των δεδομένων σε μορφή .csv για την ευκολότερη ανάγνωσή τους με το Υπολογιστικό Φύλλο Calc.

ΚΕΦΑΛΑΙΟ 6: ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

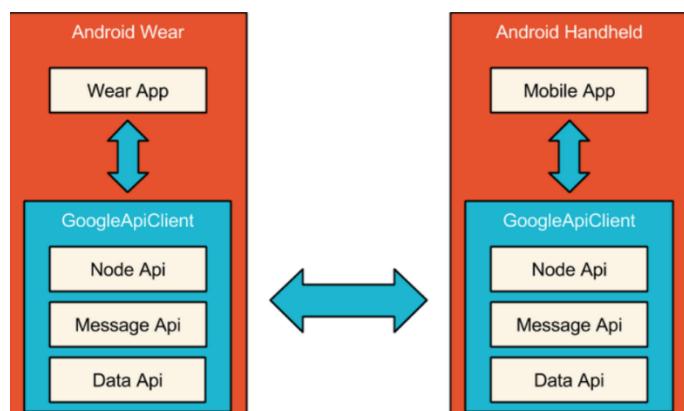
6.1 Διάρθρωση Συστήματος

Στα πλαίσια της παρούσης διπλωματικής εργασίας αναπτύχθηκαν δύο εφαρμογές Android οι οποίες βρίσκονται αποθηκευμένες στον ίδιο φάκελο ως ενιαίο Project του Android Studio.

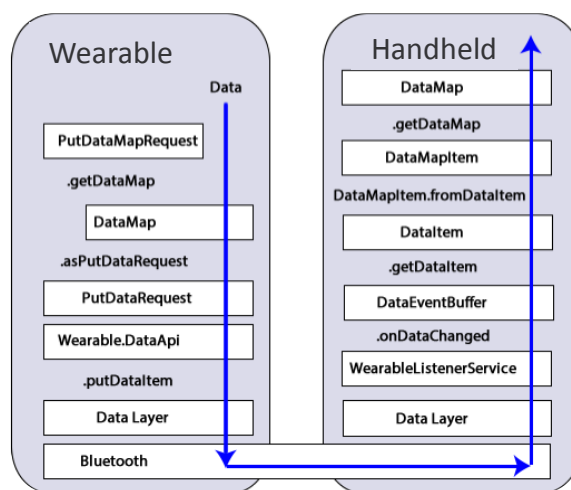
Η πρώτη είναι για τη συσκευή φορητής υπολογιστικής ικανότητας για την ανίχνευση των κινήσεων του καρπού στο χώρο με το επιταχυνσιόμετρο τριών αξόνων που διαθέτει η συσκευή και την αποστολή των δεδομένων μέσω Bluetooth στη συσκευή κινητού υπολογισμού. Το γραφικό περιβάλλον της είναι απλό και αποτελείται από ένα μόνο πεδίο κειμένου (TextField) που ενημερώνεται ανάλογα με την επιτυχία ή αποτυχία της αποστολής των δεδομένων στο μονοπάτι που έχει οριστεί.

Η βασική εφαρμογή του συστήματος η οποία αναλαμβάνει την πιο «βαριά» υπολογιστική διαδικασία είναι αυτή που αναπτύχθηκε στη συσκευή κινητού υπολογισμού η οποία μετά την επικοινωνία με το μονοπάτι στο οποίο στέλνει τα δεδομένα η συσκευή φορητής υπολογιστικής ικανότητας μέσω Bluetooth, λαμβάνει τα δεδομένα του επιταχυνσιόμετρου μετά από προτροπή του χρήστη μέσω ενός κουμπιού στη διεπαφή της εφαρμογής και στη συνέχεια με νέα επιλογή του χρήστη αναγνωρίζεται η εισερχόμενη χειρονομία με τους αλγόριθμους που διαθέτει η εφαρμογή.

Για την επικοινωνία των δύο συσκευών χρησιμοποιήθηκε το Wearable DataLayer API της Google που προσφέρει ένα αξιόπιστο μονοπάτι επικοινωνίας ανάμεσα στις συσκευές με το Data API εφόσον οι τελευταίες έχουν ήδη συζευχθεί. Η λειτουργία του συνοψίζεται στις εικόνες που ακολουθούν.



EIKONA 6.1.1: Wearable Data Layer



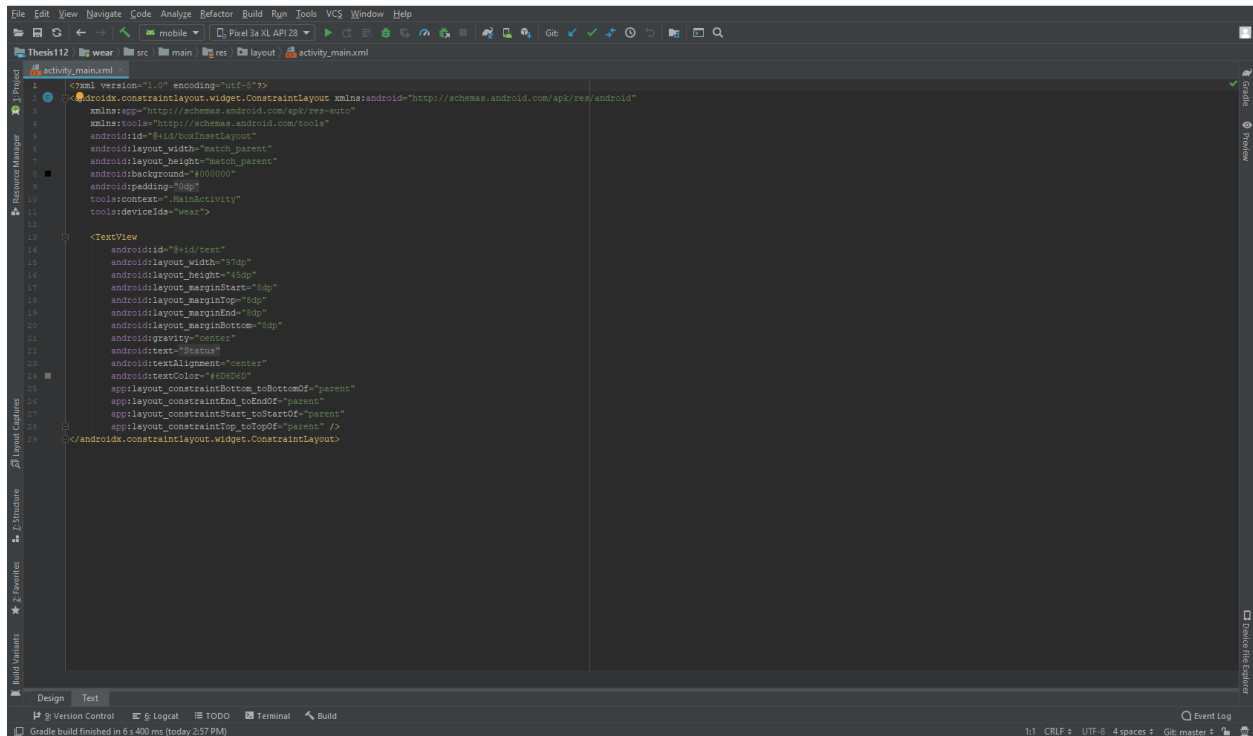
EIKONA 6.1.2: Data API

Στις εικόνες 6.1.1 και 6.1.2 βλέπουμε αναλυτικά τον τρόπο που αναγνωρίζουν οι συσκευές Android η μία την άλλη και ανταλλάσσουν δεδομένα. Με το Data API τα δεδομένα περνούν από μια διαδικασία wrapping όπου εισέρχονται σε ένα αντικείμενο τύπου DataMap. Στη συνέχεια αφού οριστούν ως DataRequest αποθηκεύονται στο μονοπάτι του Data Layer. Όταν υπάρχει διαθέσιμη συσκευή συνδεδεμένη στο ίδιο μονοπάτι μέσω Bluetooth τότε τα δεδομένα αποστέλλονται και ακολουθεί μια αντίστροφη διαδικασία για unwrapping ώστε να ληφθεί η αρχική πληροφορία. Στη συγκεκριμένη υλοποίηση προστέθηκε μια εντολή μετά το βήμα PutDataRequest όπου το πακέτο δεδομένων ορίστηκε ως setUrgent ώστε να έχει προτεραιότητα ανάμεσα στα υπόλοιπα δεδομένα που ανταλλάσσει το smart watch με το smart phone και να ξεκινάει άμεσα η αποστολή του για την αποφυγή απώλειας πληροφορίας.

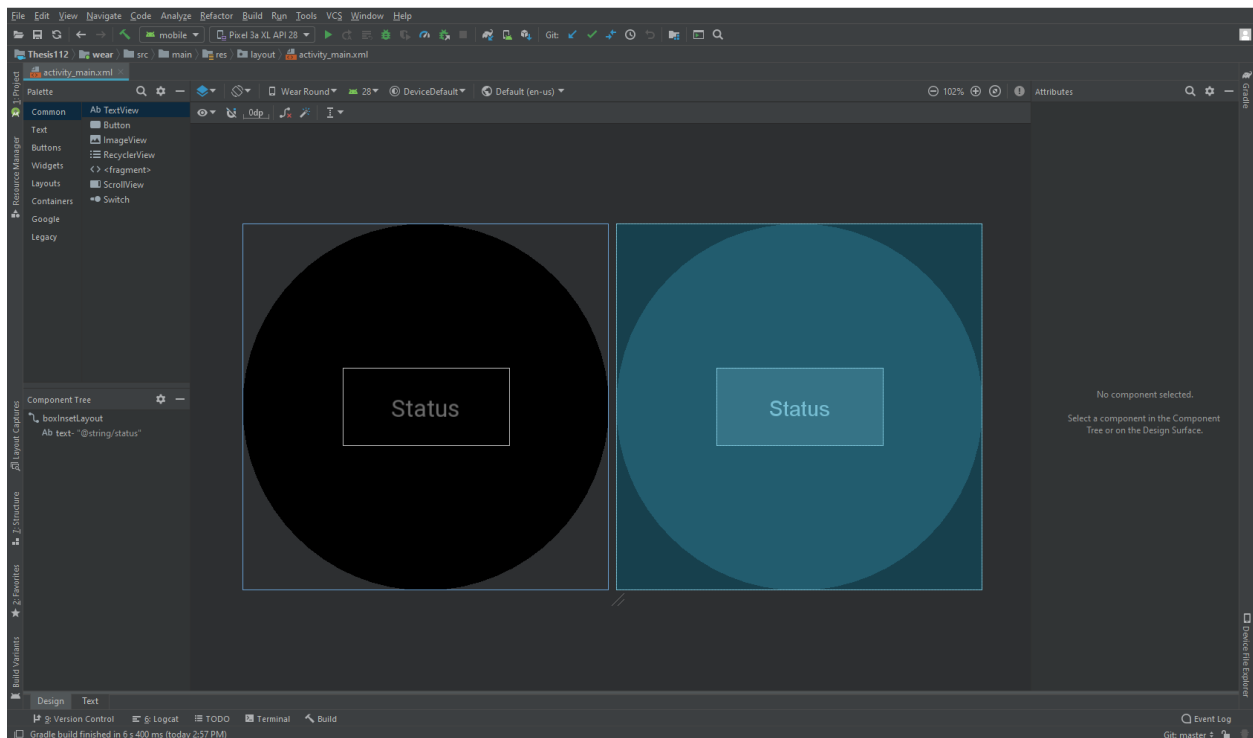
Η ανάπτυξη του κώδικα και για τις δύο εφαρμογές έγινε εξ' ολοκλήρου στο Android Studio 3.5. Ο κώδικας όλου του συστήματος, εκτενώς σχολιασμένος, καθώς και τα απαραίτητα αρχεία που χρειάζεται η εφαρμογή για να λειτουργήσει βρίσκονται στο repository του github στη διεύθυνση: <https://github.com/molarity69/Thesis111>

6.1.1 Ανάπτυξη Εφαρμογής Έξυπνου Ρολογιού

Συνηθίζεται στην ανάπτυξη της επικοινωνίας μεταξύ δύο συσκευών το σύστημα να χωρίζεται στους ρόλους master και slave κυρίως για λόγους συγχρονισμού. Σε αυτή την προσέγγιση δεν ακολουθήθηκε αυτό το μοντέλο ανάπτυξης καθώς για λόγους ακρίβειας ο στόχος ήταν η άμεση αποστολή των δεδομένων από το επιταχυνσιόμετρο κατά τη στιγμή της ανίχνευσής τους με σκοπό να λάβει τα απαραίτητα δεδομένα η εφαρμογή του έξυπνου τηλεφώνου μόνο μετά από εντολή του χρήστη. Συνεπώς η εφαρμογή του έξυπνου ρολογιού μοιάζει περισσότερο με beacon που στέλνει διαρκώς δεδομένα στο μονοπάτι της επικοινωνίας. Το πεδίο κειμένου στη διεπαφή της εφαρμογής που αναφέρθηκε παραπάνω είναι γραμμένο σε XML αν και το Android Studio παρέχει τη δυνατότητα πρόσθεσης πεδίων με drag&drop από την παλέτα του περιβάλλοντος ανάπτυξης. (Εικόνα 6.1.1.1 και Εικόνα 6.1.1.2)

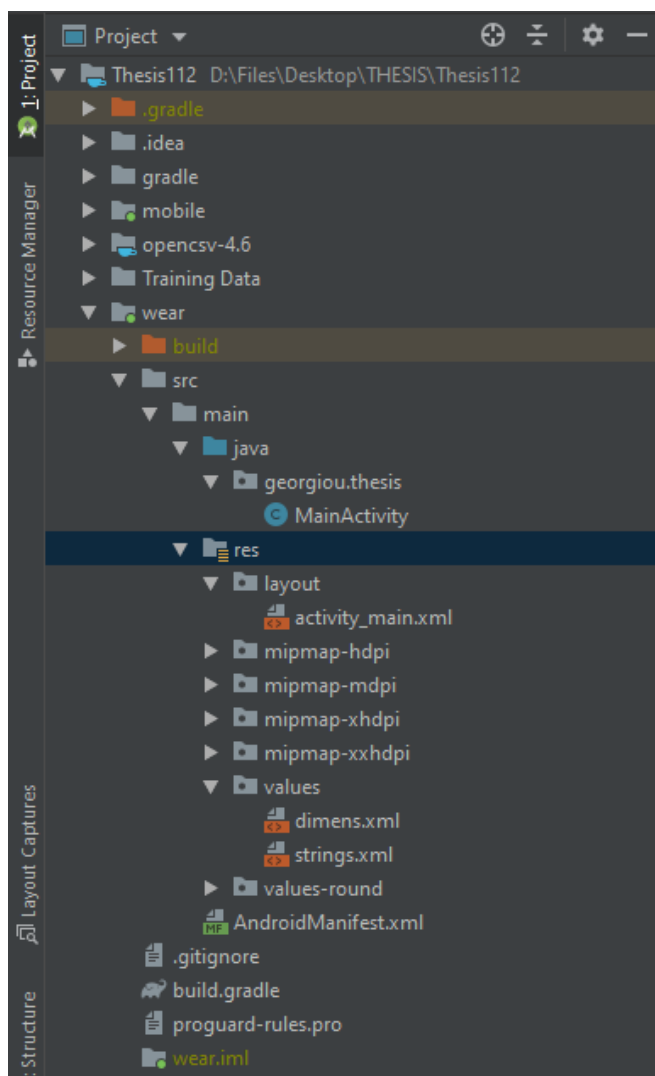


EIKONA 6.1.1.1: XML Text Editor



EIKONA 6.1.1.2: XML Designer

Η δομή των αρχείων στον φάκελο wear του project της εφαρμογής είναι αυτή που φαίνεται στην εικόνα που ακολουθεί.



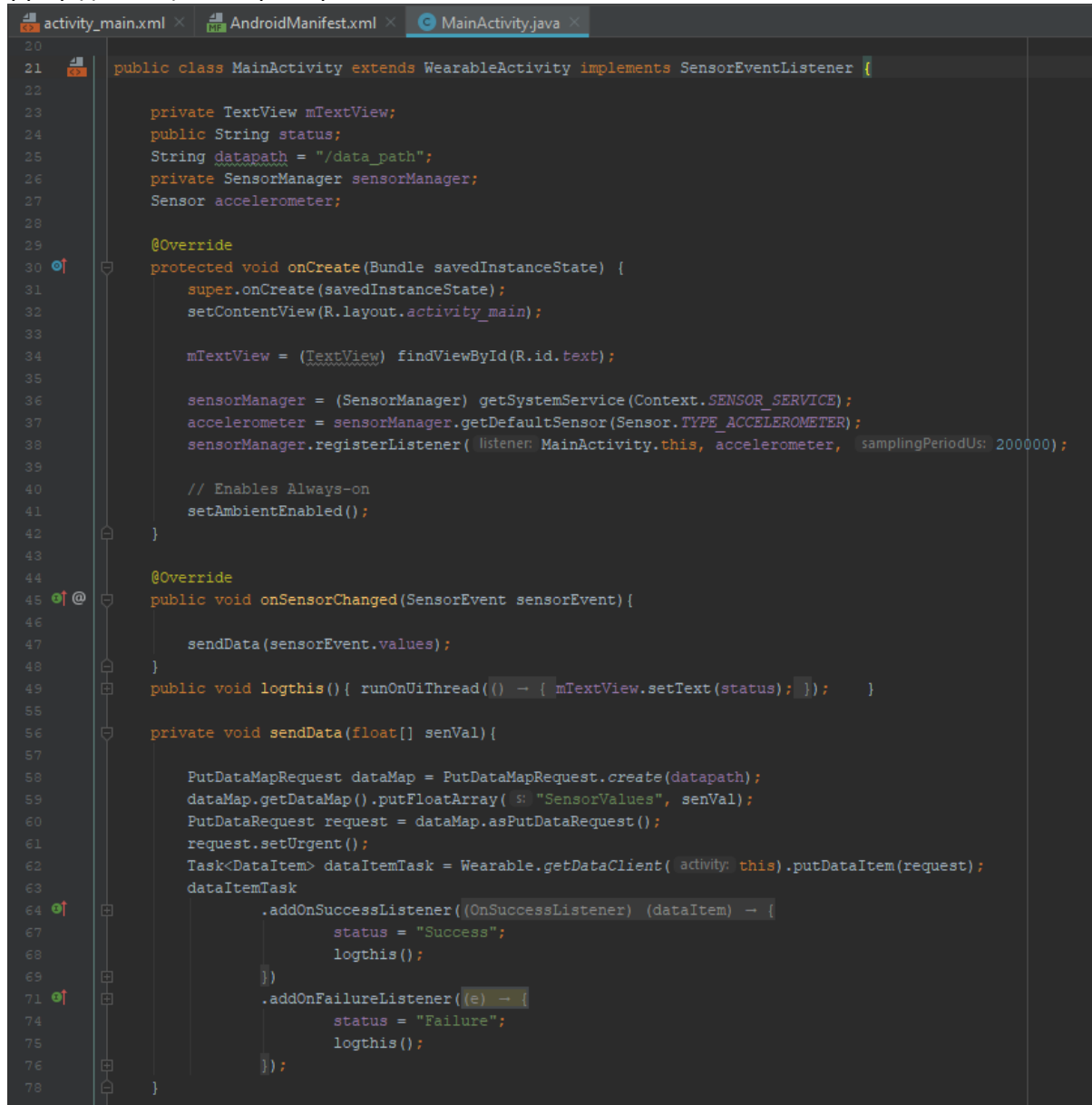
ΕΙΚΟΝΑ 6.1.1.2: Δομή Αρχείων Wear Module

Το αρχείο `activity_main.xml` που βρίσκεται στο μονοπάτι `src.main.res.layout` περιέχει τον κώδικα XML που σχηματίζει τη διεπαφή της εφαρμογής όπως φαίνεται στις παραπάνω εικόνες (Εικόνα 6.1.1.1 και Εικόνα 6.1.1.2).

Στο μονοπάτι `wear.src.main.java.georgiou.thesis` βρίσκεται το αρχείο `MainActivity.java` το οποίο περιέχει τον κώδικα που αρχικοποιεί τον αισθητήρα του επιταχυνσιόμετρου και ορίζει το μονοπάτι επικοινωνίας με το έξυπνο τηλέφωνο. (Εικόνα 6.1.1.3)

Απαραίτητη για τη σωστή λειτουργία της εφαρμογής είναι η δήλωση της `MainActivity` του wear module στο αρχείο `AndroidManifest.xml`.

Στην εικόνα που ακολουθεί φαίνεται ο κώδικας με τις κυριότερες λειτουργίες της εφαρμογής του έξυπνου ρολογιού.



```
20
21 public class MainActivity extends WearableActivity implements SensorEventListener {
22
23     private TextView mTextView;
24     public String status;
25     String datapath = "/data_path";
26     private SensorManager sensorManager;
27     Sensor accelerometer;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_main);
33
34         mTextView = (TextView) findViewById(R.id.text);
35
36         sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
37         accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
38         sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_NORMAL);
39
40         // Enables Always-on
41         setAmbientEnabled();
42     }
43
44     @Override
45     public void onSensorChanged(SensorEvent sensorEvent) {
46         sendData(sensorEvent.values);
47     }
48
49     public void logthis() { runOnUiThread(() -> { mTextView.setText(status); }); }
50
51     private void sendData(float[] senVal) {
52
53         PutDataMapRequest dataMap = PutDataMapRequest.create(datapath);
54         dataMap.getDataMap().putFloatArray("SensorValues", senVal);
55         PutDataRequest request = dataMap.asPutDataRequest();
56         request.setUrgent();
57         Task<DataItem> dataItemTask = Wearable.getDataClient(this).putDataItem(request);
58         dataItemTask
59             .addOnSuccessListener((OnSuccessListener) (dataItem) -> {
60                 status = "Success";
61                 logthis();
62             })
63             .addOnFailureListener((e) -> {
64                 status = "Failure";
65                 logthis();
66             });
67     }
68 }
```

ΕΙΚΟΝΑ 6.1.1.3: Ο κώδικας του Wear Module

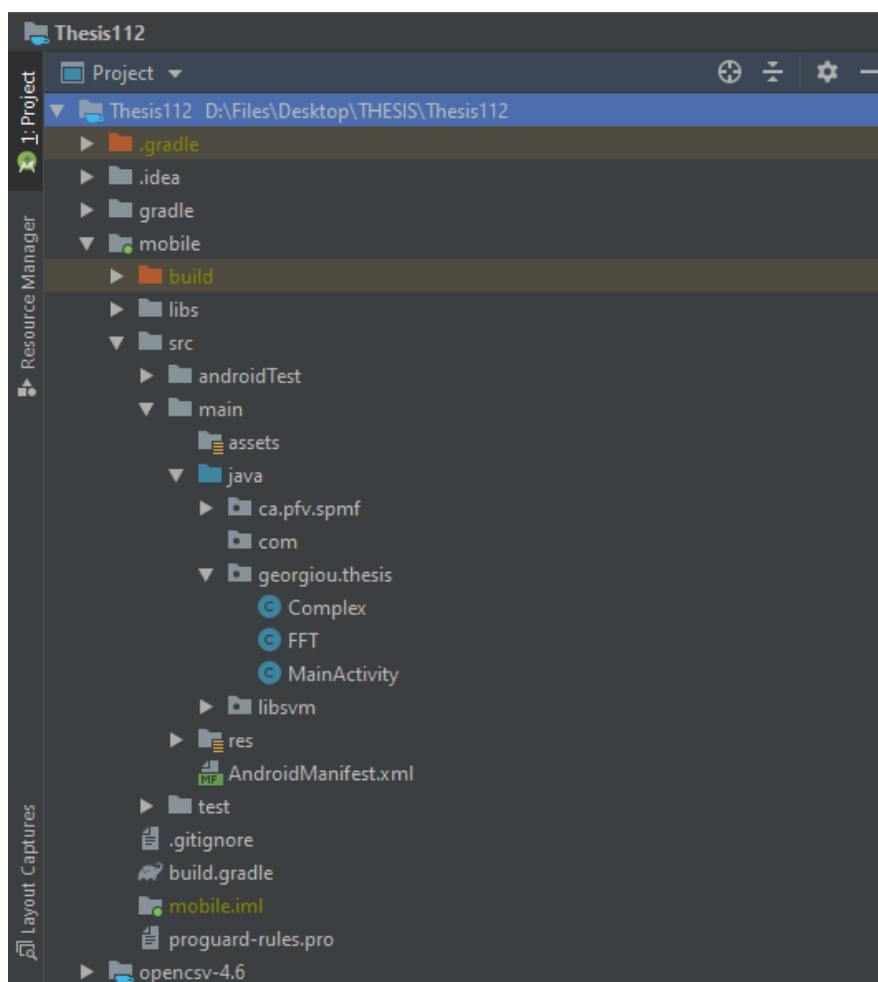
Η μεταβλητή datapath τύπου String που ορίζεται στη γραμμή 25 του κώδικα είναι το μονοπάτι στο κανάλι του Bluetooth που επικοινωνεί το έξυπνο ρολόι με το συζευγμένο έξυπνο κινητό το οποίο χρειάζεται να δηλωθεί για να σταλούν τα δεδομένα του επιταχυνσιόμετρου. Το ίδιο μονοπάτι ορίζεται και στον κώδικα της εφαρμογής του έξυπνου κινητού. Η αποστολή των δεδομένων γίνεται μέσω της μεθόδου sendData (γραμμή 56), που αναλαμβάνει το wrapping των δεδομένων σύμφωνα με το API της Google, κάθε φορά που καλείται η μέθοδος onSensorChanged (ανά 200 millisecond όπως ορίζεται στη γραμμή 38) που παρέχει το αντικείμενο τύπου Sensor της κλάσης SensorManager (γραμμή 45).

6.1.2 Ανάπτυξη Εφαρμογής Συσκευής Κινητού Υπολογισμού

Η εφαρμογή στη συσκευή κινητού υπολογισμού θα μπορούσε να χαρακτηριστεί ως ο «εγκέφαλος» του συστήματος καθώς όλες οι λειτουργίες διαχείρισης δεδομένων και αναγνώρισης των χειρονομιών εξαρτώνται από αυτή. Σε αυτή την υποενότητα, αναλύεται η βασική λειτουργία της επικοινωνίας των δύο συσκευών από την πλευρά του έξυπνου τηλεφώνου και η διαχείριση δεδομένων. Για τις υπόλοιπες λειτουργίες της εφαρμογής ακολουθεί αναλυτική επεξήγηση στις παρακάτω υποενότητες.

Σε αντίθεση με τη διεπαφή του έξυπνου ρολογιού, η διεπαφή της συσκευής κινητού υπολογισμού δημιουργήθηκε με το εργαλείο XML Designer που παρέχει το Android Studio για λόγους ταχύτητας με μόνη εξαίρεση την υλοποίηση του popup menu για την επιλογή του data set που θα χρησιμοποιηθεί στην αναγνώριση της χειρονομίας με τον αλγόριθμο SAX.

Στην εικόνα που ακολουθεί φαίνεται η δομή των αρχείων του υποσυστήματος της συσκευής κινητού υπολογισμού. (Εικόνα 6.1.2.1).



ΕΙΚΟΝΑ 6.1.2.1: Δομή Αρχείων Mobile Module

Στο φάκελο `main.java` περιέχονται όλα τα αρχεία κώδικα που χρησιμοποιεί η εφαρμογή χωρισμένα σε `packages` για την εύκολη πρόσβαση και τροποποίησή τους ανάλογα με τις ανάγκες της εφαρμογής.

- Φάκελος **ca.pfv.spmf**

Ο φάκελος `ca.pfv.spmf` περιέχει όλα τα αρχεία της βιβλιοθήκης SPMF που χρησιμοποιούνται για την αναπαράσταση των χρονοσειρών με τη μέθοδο SAX.

- Φάκελος **georgiou.thesis**

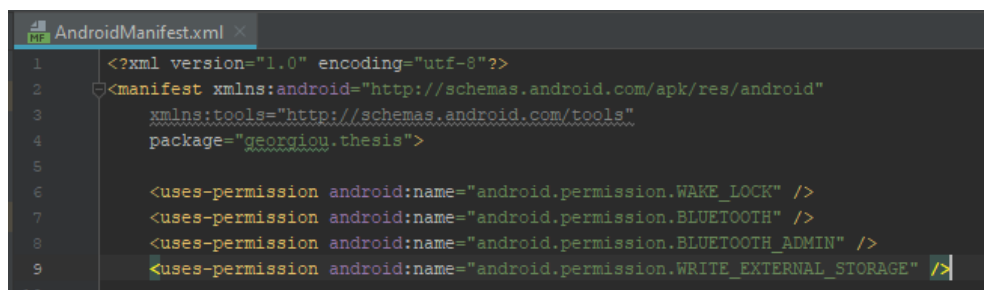
Στο φάκελο `georgiou.thesis` περιέχονται τα αρχεία κώδικα `Complex.java` και `FFT.java` που είναι υπεύθυνα για τη μετατροπή των δεδομένων σε μιγαδικούς αριθμούς και το μετασχηματισμό τους με τον FFT, αντίστοιχα όπως και το αρχείο `MainActivity.java` που περιέχει όλο τον κώδικα που συντονίζει τη λειτουργία της εφαρμογής.

- Φάκελος **libsvm**

Τέλος, στο φάκελο `libsvm` βρίσκονται όλα τα αρχεία που χρησιμοποιεί η βιβλιοθήκη `libsvm` για τη σωστή λειτουργία του SVM που χρησιμοποιείται στην αναγνώριση της εισερχόμενης χειρονομίας με τη μέθοδο FFT.

Ο φάκελος `res` περιέχει τα αρχεία κώδικα που απαρτίζουν τη διεπαφή της εφαρμογής όπως και τα αρχεία ήχου που καλούνται για αναπαραγωγή από τον κώδικα της εφαρμογής κατά την αναγνώριση της χειρονομίας ως παράδειγμα για την ανάθεση μιας λειτουργίας στο σύστημα μετά την αναγνώριση.

Απαραίτητη για τη λειτουργία της εφαρμογής είναι η δήλωση των λειτουργιών του Bluetooth (σειρές 7 και 8) και της προσπέλασης των αρχείων της συσκευής (σειρά 9) στο αρχείο `AndroidManifest.xml` όπως φαίνεται στην ακόλουθη εικόνα.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="georgiou.thesis">
5
6     <uses-permission android:name="android.permission.WAKE_LOCK" />
7     <uses-permission android:name="android.permission.BLUETOOTH" />
8     <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
9     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

ΕΙΚΟΝΑ 6.1.2.2: Δήλωση λειτουργιών της εφαρμογής στο αρχείο `AndroidManifest.xml`

Για τη λήψη των δεδομένων, από το μονοπάτι που τα στέλνει το έξυπνο ρολόι, οι βασικές λειτουργίες που έπρεπε να υλοποιηθούν ήταν οι εξής.

Η δήλωση ενός Bluetooth Adapter για τη σύνδεση στο μονοπάτι που στέλνει τα δεδομένα του επιταχυνσιόμετρου ή εφαρμογή του έξυπνου ρολογιού (Εικόνα 6.1.2.3). Επίσης στην έναρξη της εφαρμογής μαζί με την αρχικοποίηση του Listener γίνεται και έλεγχος για το αν ο χρήστης έχει δώσει δικαιώματα πρόσβασης στον αποθηκευτικό χώρο της συσκευής ώστε να του εμφανίσει το κατάλληλο μήνυμα. (Εικόνα 6.1.2.4) Σειρά έχει η προσπέλαση των εισερχόμενων δεδομένων με τη μέθοδο του Listener, `onDataChanged()` (Εικόνα 6.1.2.5). Στη συνέχεια υλοποιήθηκε μια μέθοδος που αποθηκεύει τα δεδομένα σε μία λίστα για την περαιτέρω επεξεργασία τους μέσω των αλγορίθμων ή την εξαγωγή τους σε αρχεία (Εικόνα 6.1.2.6).

```
85
86     public BluetoothAdapter mBluetoothAdapter; //bluetooth initializer
87     public String datapath = "/data_path"; //path for bluetooth communication
88
```

ΕΙΚΟΝΑ 6.1.2.3: Δήλωση του BluetoothAdapter και του datapath

```
239
240     /* register bluetooth adapter and listener */
241     mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
242     Wearable.getDataClient( activity: this).addListener( onDataChangedListener: this);
243
244     /* prompt user to grant storage permission for I/O */
245     if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED) {
246         ActivityCompat.requestPermissions( activity: this, new String[] {Manifest.permission.WRITE_EXTERNAL_STORAGE}, requestCode: 0);
247     }
248
```

ΕΙΚΟΝΑ 6.1.2.4: Αρχικοποίηση του BluetoothAdapter και του Listener. Έλεγχος για δικαιώματα εγγραφής στη συσκευή.

```

497 override
498 public void onDataChanged (@NonNull DataEventBuffer dataEventBuffer) {
499
500     for (DataEvent event : dataEventBuffer) { //for every incoming event
501         if (event.getType() == DataEvent.TYPE_CHANGED) { //if there is a data change
502             String path = event.getDataItem().getUri().getPath(); //get the path
503             if (path.equals(datapath)) { //if path equals the path to our smart watch
504                 DataMapItem dataMapItem = DataMapItem.fromDataItem(event.getDataItem()); //unwrap the item
505                 values = dataMapItem.getDataMap().getFloatArray( "SensorValues"); //get the contents into the array
506                 if(values!=null && start) { //if the array has data and the button to start recording has been pressed
507                     //this if-statement checks when the incoming values number reaches 30
508                     //it's written this way so that the logthis function can update the total recorded number on the UI
509                     //and knowing when to stop recording
510                     if((recordedCount - gesturesComplete) % 31 == 0){
511                         Toast.makeText( context, this, text: "Recording completed", Toast.LENGTH_LONG).show();
512                         start = false; //stop recording the incoming data in onDataChanged method
513                         recordedCount++; //increment recordings by one in order to make the above if-statement work
514                         gesturesComplete++; //increment the gestures completed by one to keep track
515                     }
516                     else { //while recording each incoming data packet
517                         dataBuffer(); //call the method that stores the incoming data in the arrayList
518                         logthis(); //update the UI
519                     }
520                 }
521                 else{
522                     if(!start) { //if start button hasn't been pressed
523                         logthis(); //update the UI about the received but not recorded data
524                         d3Button.setEnabled(true); //keep the buttons enabled when not recording
525                         saxButton.setEnabled(true);
526                         fftButton.setEnabled(true);
527                         trainButton.setEnabled(true);
528                         //enable the RECORD A GESTURE button only when received data has reached 100
529                         //it's necessary in order to establish a stable bluetooth connection after the app has started
530                         if(receivedCount > 100) newGesture.setEnabled(true);
531                     }
532                     else Log.d(TAG, msg: "onDataChanged: Let's hope I won't be printed"); //debug info
533                 }
534             } else {
535                 Log.e(TAG, msg: "Unrecognized path: " + path); //debug info
536             }
537         } else if (event.getType() == DataEvent.TYPE_DELETED) {
538             Log.v(TAG, msg: "Data deleted : " + event.getDataItem().toString()); //debug info
539         } else {
540             Log.e(TAG, msg: "Unknown data event Type = " + event.getType()); //debug info
541         }
542     }
543     receivedCount++; //increment the received packets count
544 }

```

ΕΙΚΟΝΑ 6.1.2.5: Μέθοδος `onDataChanged` για τη λήψη των δεδομένων από το έξυπνο ρολόι

Η μέθοδος `onDataChanged` του Listener που φαίνεται στην παραπάνω εικόνα καλείται κάθε φορά που ανιχνεύεται αλλαγή των δεδομένων στο μονοπάτι επικοινωνίας μέσω Bluetooth με το έξυπνο ρολόι. Όταν ανιχνευθεί ροή δεδομένων στο μονοπάτι που έχει δηλωθεί από πριν, ακολουθεί `unwrapping` των δεδομένων και η αποθήκευσή τους σε έναν πίνακα τριών θέσεων, μια θέση για κάθε συνιστώσα του επιταχυνσιόμετρου, με το όνομα `values`. Στη συνέχεια αν ο πίνακας περιέχει τα δεδομένα και ο χρήστης έχει πιέσει το κουμπί της καταγραφής, καλείται η μέθοδος `dataBuffer` που αποθηκεύει τα δεδομένα σε μια λίστα μέχρι το πλήθος τους να φτάσει τα 30, όπως καλείται και η μέθοδος `logthis` που ενημερώνει ένα πεδίο του UI για το πλήθος των εισερχόμενων δεδομένων. Στο τέλος της καταγραφής των δεδομένων ο χρήστης ενημερώνεται για την ολοκλήρωση της διαδικασίας με την εμφάνιση ενός μηνύματος `Toast` και ενεργοποιούνται τα κουμπιά της διεπαφής τα οποία προηγουμένως είχαν απενεργοποιηθεί με το πάτημα του κουμπιού της έναρξης της καταγραφής για να αποφευχθούν ακούσια σφάλματα.

```

546  /** dataBuffer is the method that when called stores the incoming data in an ArrayList */
547  public void dataBuffer() {
548
549      gestureValuesBufferX.add(values[0]); //add the next X value int the List
550      gestureValuesBufferY.add(values[1]); //add the next Y value int the List
551      gestureValuesBufferZ.add(values[2]); //add the next Z value int the List
552
553      recordedCount++; //increment the recorded count by one
554
555      if(gestureValuesBufferX.size() == 30 && gestureValuesBufferY.size() == 30 && gestureValuesBufferZ.size() == 30) { //when all lists are full
556
557          if(gestureGeneralBuffer.size() >= 90) { //check if there is any data left in the buffer list from a previous call in case I forgot to clear it
558              gestureGeneralBuffer.clear(); //and clear it
559          }
560
561          gestureGeneralBuffer.addAll(0, gestureValuesBufferX); //copy the X values into the first 30 positions
562          gestureGeneralBuffer.addAll(30, gestureValuesBufferY); //copy the Y values into the next 30 positions
563          gestureGeneralBuffer.addAll(60, gestureValuesBufferZ); //copy the Z values into the last 30 positions
564          gestureValuesBufferX.clear(); //then clear all the small buffers
565          gestureValuesBufferY.clear();
566          gestureValuesBufferZ.clear();
567      }
568  }

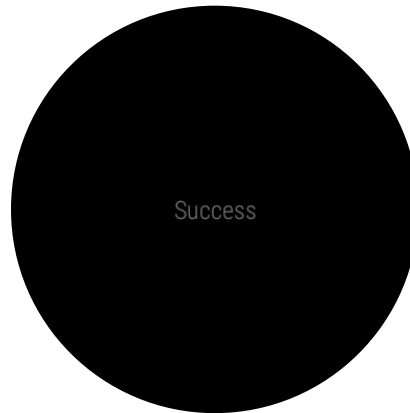
```

ΕΙΚΟΝΑ 6.1.2.6: Μέθοδος *dataBuffer* για την αποθήκευση των εισερχόμενων δεδομένων σε λίστα.

Η μέθοδος *dataBuffer* καλείται με την άφιξη κάθε νέας τριάδας δεδομένων X, Y, Z από το επιταχυνσιόμετρο μέσω του Bluetooth μέσω της μεθόδου *onDataChanged* που αναλύθηκε παραπάνω. Σε κάθε κλήση της, κάθε μία τιμή εκ των συνιστωσών που περιέχει ο πίνακας *values* που αναφέρθηκε προηγουμένως, αποθηκεύεται σε μία λίστα. Στη συνέχεια ακολουθεί ένας έλεγχος για το μέγεθος των λιστών και αν το πλήθος των θέσεων των τελευταίων έχει φτάσει τις 30, σβήνονται τα προηγούμενα περιεχόμενα της γενικής λίστας που περιέχει και τις 90 τιμές του επιταχυνσιόμετρου ($30x + 30y + 30z$) σε περίπτωση που περιέχει δεδομένα από προηγούμενη κλήση. Στο τέλος όλες οι τιμές της λίστας με τα δεδομένα της συνιστώσας X αποθηκεύονται στις πρώτες 30 θέσεις της γενικής λίστας, της συνιστώσας Y στις επόμενες 30 θέσεις και της Z στις τελευταίες 30 θέσεις ενώ οι λίστες της κάθε συνιστώσας αδειάζουν για να είναι έτοιμες για την επόμενη χειρονομία. Η διαδικασία αυτή ακολουθείται με το συγκεκριμένο τρόπο για να καταλήξουν τα δεδομένα της χειρονομίας σε έναν πίνακα και να αποτελούν μια χρονοσειρά αντί για τρεις καθιστώντας ευκολότερη τη διαχείρισή τους με τον εκάστοτε αλγόριθμο.

6.2 Γραφικό Περιβάλλον και Πλοήγηση

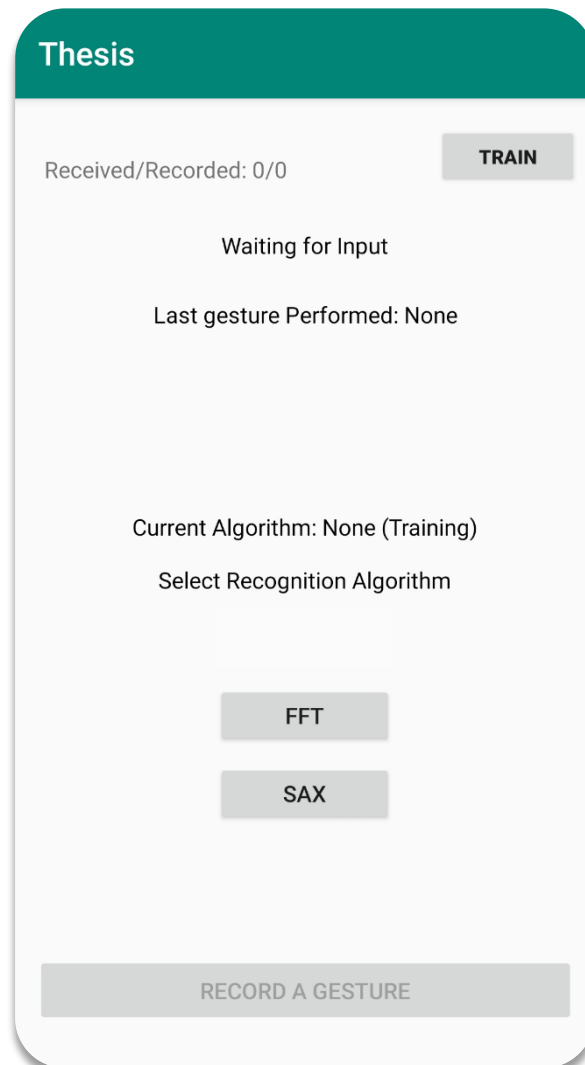
Το γραφικό περιβάλλον της συσκευής φορητής υπολογιστικής ικανότητας αποτελείται από ένα μόνο πεδίο κειμένου που κατά την έναρξη της εφαρμογής έχει τη μορφή που φαίνεται στον XML Designer (Εικόνα 6.1.1.2) και μετά την επιτυχή αποστολή των δεδομένων στο μονοπάτι που έχει οριστεί το πεδίο ενημερώνεται και παίρνει τη μορφή που φαίνεται στην εικόνα που ακολουθεί.



ΕΙΚΟΝΑ 6.2.1: Διεπαφή Έξυπνου Ρολογιού

Σε περίπτωση αποτυχίας αποστολής των δεδομένων το πεδίο ενημερώνεται με τη λέξη "Failure". Δε διατίθεται το αντίστοιχο screenshot καθώς σε καμία από τις δοκιμές της εφαρμογής δεν απέτυχε η αποστολή των δεδομένων.

Για την ευκολία στο χειρισμό όλων των λειτουργιών που παρέχει η εφαρμογή της συσκευής κινητού υπολογισμού έγινε ανάπτυξη της διεπαφής που τα σενάρια χρήσης της περιγράφονται στις παρακάτω εικόνες (Εικόνα 6.2.2, Εικόνα 6.2.3, Εικόνα 6.2.4, Εικόνα 6.2.5).



ΕΙΚΟΝΑ 6.2.2: Η μορφή της διεπαφής κατά την έναρξή της

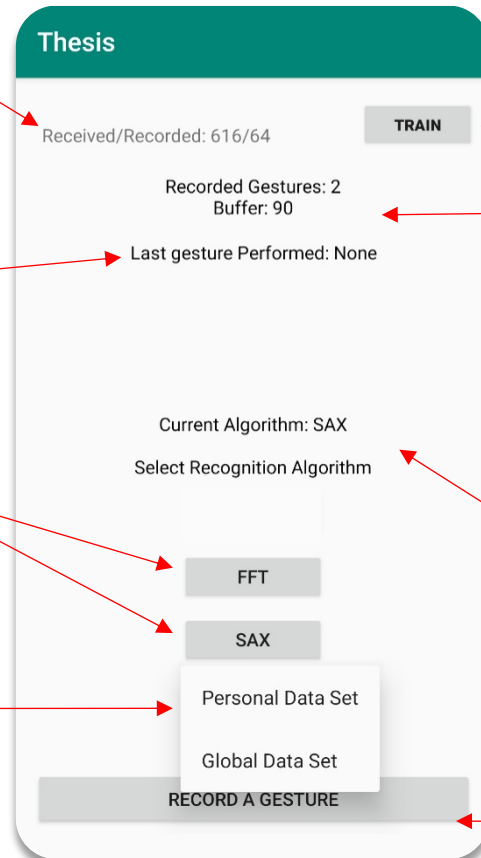
Το κουμπί "RECORD A GESTURE" παραμένει απενεργοποιημένο μέχρι το πλήθος των πακέτων που έχουν ληφθεί να ξεπεράσει τα 100 ώστε να σταθεροποιηθεί ο ρυθμός των εισερχόμενων δεδομένων εφόσον παρατηρήθηκαν αυξομειώσεις στην ταχύτητα λήψης των δεδομένων μέχρι αυτό το όριο.

Πεδίο που ενημερώνει για τη λήψη και την αποθήκευση των πακέτων δεδομένων.

Πεδίο που εμφανίζει ποια χειρονομία έχει εκτελέσει τελευταία ο χρήστης.

Κουμπιά επιλογής του αντίστοιχου αλγόριθμου για την αναγνώριση της εισερχόμενης χειρονομίας.

Menu που επιλέγει ο χρήστης με ποιο data set να γίνει η αναγνώριση.



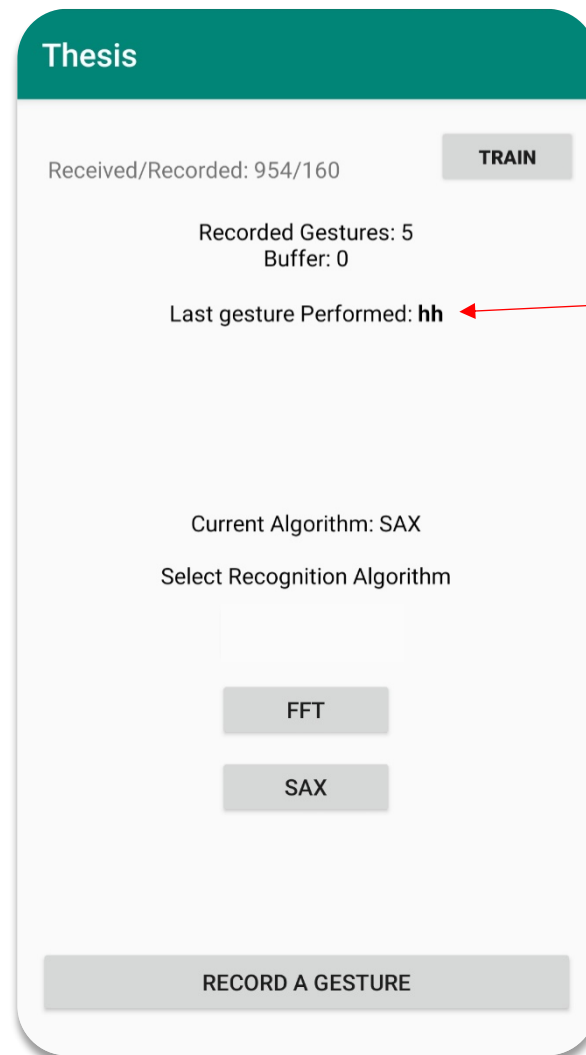
Κουμπί για την αποθήκευση χειρονομιών.

Πεδίο που ενημερώνει για το σύνολο των καταγεγραμμένων χειρονομιών και την κατάσταση του buffer με τα εισερχόμενα δεδομένα της χειρονομίας.

Πεδίο που εμφανίζει ποιος αλγόριθμος έχει επιλεγεί για αναγνώριση.

Κουμπί που ξεκινάει να καταγράφει τα εισερχόμενα δεδομένα της χειρονομίας.

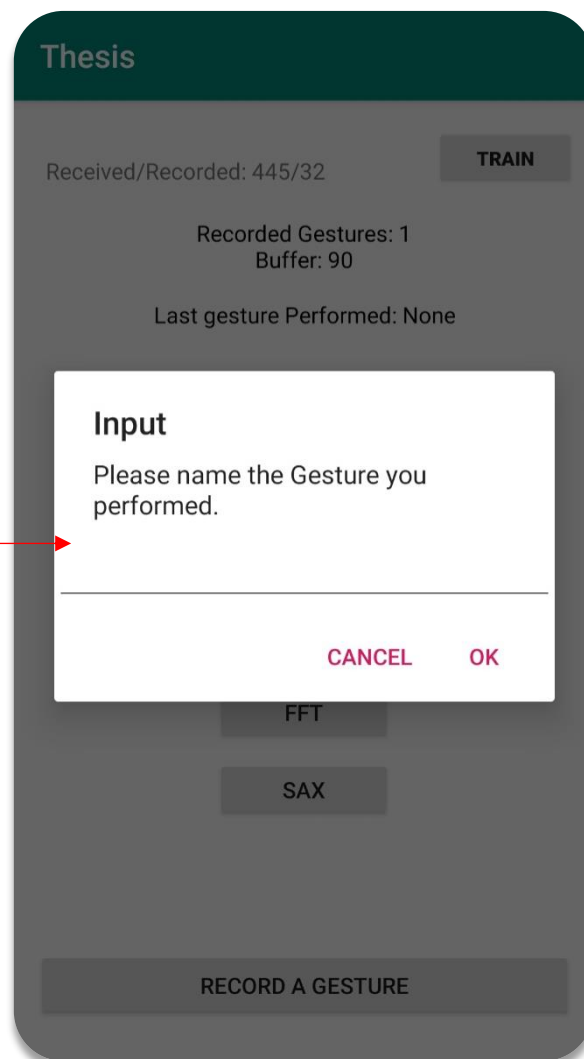
ΕΙΚΟΝΑ 6.2.3: Η μορφή της διεπαφής κατά τη διαδικασία επιλογής αναγνώρισης με τον αλγόριθμο SAX



Στο πεδίο εμφανίστηκε η χειρονομία που αναγνωρίστηκε με τον αλγόριθμο SAX μετά την εκτέλεση.

ΕΙΚΟΝΑ 6.2.4: Η μορφή της διεπαφής μετά την αναγνώριση της χειρονομίας

Όταν ο χρήστης πιάσει το κουμπί TRAIN εμφανίζεται το PopUp Dialog που τον προτρέπει να ονομάσει τη χειρονομία που εκτέλεσε για να αποθηκευτεί.



ΕΙΚΟΝΑ 6.2.5: Η μορφή της διεπαφής όταν ο χρήστης πιάσει το κουμπί *TRAIN*

6.3 Εξαγωγή Δεδομένων

Μετά την ολοκλήρωση της εκτέλεσης μιας χειρονομίας όπως φαίνεται στις παραπάνω εικόνες και εφόσον ο χρήστης επιλέξει να αποθηκεύσει τη χειρονομία που εκτέλεσε, εισάγει το όνομα της στο αναδυόμενο παράθυρο και στη συνέχεια η χειρονομία προσαρτάται ως τελευταία εγγραφή σε ένα αρχείο τύπου .csv (Comma Separated Value) με τον ακόλουθο κώδικα (Εικόνα 6.3.1).

```
1254  /** exportDataToCSV method writes training data, complex data after FFT and absolute values of the FFT into a CSV file
1255  * @param accData The data to be exported
1256  * @param fp The path to the file
1257  * @param FL The file name*/
1258
1259  @ public void exportDataToCSV(float[] accData, String fp, File FL, boolean transposed, String value){
1260
1261      try{
1262          if(FL.exists() && !FL.isDirectory()) { //if the file already exists
1263              FileWriter mFileWriter = new FileWriter(fp, append: true); //append to it
1264              writer = new CSVWriter(mFileWriter); //initialize the writer
1265          }
1266          else {
1267              writer = new CSVWriter(new FileWriter(fp)); //else initialize a writer to write a new file
1268          }
1269
1270          if(accData.length == 3) { //if the method is called with an array of length 3
1271
1272              String[] val = new String[gestureGeneralBuffer.size()+1];
1273              for(int i = 0; i < val.length-1; i++){
1274                  val[i] = Float.toString(gestureGeneralBuffer.get(i));
1275              }
1276              val[gestureGeneralBuffer.size()] = value;
1277              writer.writeNext(val, applyQuotesToAll: false);
1278          }
1279      } catch (IOException e) {
1280          e.printStackTrace(); //handle exceptions
1281      }
1282  }
```

ΕΙΚΟΝΑ 6.3.1: Μέθοδος *exportDataToCSV* που εξάγει τα δεδομένα της χειρονομίας σε ένα αρχείο τύπου *csv*.

Η μέθοδος *exportDataToCSV* καλείται με διαφορετικούς τρόπους χρησιμοποιώντας τις κατάλληλες παραμέτρους για να κάνει εξαγωγή των δεδομένων στα αντίστοιχα αρχεία που αναφέρθηκαν στο κεφάλαιο 4.3 Δομή Αρχείων Αποθήκευσης.

Ο τρόπος εξαγωγής των δεδομένων στα αρχεία τύπου .txt τα οποία χρησιμοποιεί ο αλγόριθμος SAX για να μετασχηματίσει την αναπαράσταση των χρονοσειρών γίνεται με την ακόλουθη μέθοδο (Εικόνα 6.3.2).

```
696  /**
697   * writeSAXtoTXT method was used during development for exporting raw data to txt for transforming them into a new data set with SAX
698   * @param dataSet is the set you wish the function to initialize with
699   * @param f the name of the file to write to
700   */
701  public void writeSAXtoTXT(float[][] dataSet, File f){
702
703      try {
704          BufferedWriter TXTWriter = new BufferedWriter(new FileWriter(f)); //initialize writer
705          for(float[] valsRow : dataSet){ //for all the dataset array
706              for(float valsCol : valsRow) {
707                  TXTWriter.append(Float.toString(valsCol)); //append every value to the text file
708                  TXTWriter.append(","); //separate it with coma
709              }
710              TXTWriter.newLine(); //change line
711          }
712          TXTWriter.flush(); //flush the writer
713          TXTWriter.close(); //close the writer
714      }catch (Exception e){
715          e.printStackTrace(); //exception handling
716      }
717  }
```

ΕΙΚΟΝΑ 6.3.2: Μέθοδος writeSAXtoTXT που εξάγει τα δεδομένα των αποθηκευμένων χειρονομιών σε αρχείο txt.

Σε αντίθεση με τη μέθοδο exportDataToCSV που προσαρτεί κάθε νέα εγγραφή στην τελευταία σειρά του αρχείου, η μέθοδος writeSAXtoTXT δημιουργεί από την αρχή το κάθε αρχείο καθώς με κάθε νέα χειρονομία που αποθηκεύεται γίνεται νέα αρχικοποίηση των πινάκων με τις αποθηκευμένες χειρονομίες που παίρνει με την παράμετρο dataSet.

6.4 Αλγόριθμος FFT

Μετά από έρευνα στη σχετική βιβλιογραφία και με τη μελέτη των εργασιών που αναφέρθηκαν στο κεφάλαιο 2.1 Σχετικές Μελέτες και Εργασίες, μία μέθοδος που υπόσχεται μεγάλο ποσοστό ακρίβειας στην αναγνώριση χειρονομιών είναι με την επεξεργασία των δεδομένων με μετασχηματισμό Fourier και κατηγοριοποίηση με την είσοδο των δεδομένων σε μια μηχανή υποστήριξης διανυσμάτων.

6.4.1 Επεξεργασία Δεδομένων

Από τη στιγμή που έχει ολοκληρωθεί η εκτέλεση της χειρονομίας από το χρήστη και οι 90 τιμές των δεδομένων περιέχονται στη λίστα που δημιουργήθηκε, οι τελευταίες περνούν από δειγματοληψία αγνοώντας κάθε τρίτη τιμή που συναντάται στη λίστα. Για αυτή τη λειτουργία υλοποιήθηκε μία μέθοδος με σκοπό τη μείωση του πλήθους των τιμών από 90 σε 64 καθώς η υλοποίηση του αλγορίθμου FFT σε Java λειτουργούσε σε πίνακες με πλήθος τιμών δυνάμεις του 2. Στη συνέχεια κάθε μία τιμή ορίζεται ως το πραγματικό μέρος ενός μιγαδικού αριθμού, με το φανταστικό μέρος να είναι μηδέν, ώστε να μπουν τα δεδομένα ως είσοδος στη μέθοδο FFT. Τέλος, εφόσον οι 64 πλέον μιγαδικές τιμές μετασχηματιστούν με τη μέθοδο FFT υπολογίζεται το μέτρο των μιγαδικών αριθμών που προκύπτουν και εξάγονται στο αρχείο **FFTAnalysisDataFloat.csv**.

6.4.2 Προσπέλαση Δεδομένων

Τα δεδομένα που χρειάζεται ο αλγόριθμος FFT για να λειτουργήσει στην εφαρμογή αρχικοποιούνται σε πίνακες κατά την έναρξη της εφαρμογής στη μέθοδο `onCreate` από το αρχείο **newClass.txt** που περιέχει τις μεταβλητές του μοντέλου SVM, όπως φαίνονται στην Εικόνα 4.3.5, το οποίο αναλαμβάνει την κατηγοριοποίηση της χειρονομίας. Το αρχείο αυτό δημιουργείται κατά την εκπαίδευση του μοντέλου του SVM και αποθηκεύεται χειροκίνητα στο φάκελο DCIM της συσκευής κινητού υπολογισμού.

Τα δεδομένα που αποθηκεύτηκαν στο αρχείο **FFTAnalysisDataFloat.csv** χρησιμοποιήθηκαν ως είσοδος για το μοντέλο του SVM που εκπαιδεύτηκε στο RStudio με το ακόλουθο Script.

The image shows the RStudio interface with a script editor open. The script contains R code for training an SVM model. It starts with removing previous data, loading necessary packages (MLmetrics, e1071, TSMining, seewave, jmotif, stringr), reading data from a CSV file, scrambling it, and then building an SVM classifier using the svm function with specific parameters (gamma = 0.02, cost = 1000). Finally, it writes the trained model to a file named newClass.txt.

```
1 #remove all previous data
2 rm(list=ls())
3
4 #load packages
5 library(MLmetrics)
6 library(e1071)
7 library(TSMining)
8 library(seewave)
9 library(jmotif)
10 library(stringr)
11
12 # Load the data from file
13 alldata = read.csv(file = "D:/Files/Desktop/THESIS/Thesis112/Training Data/FFTAnalysisDataFloat.csv")
14
15 #scramble the data set
16 alldata = alldata[sample(nrow(alldata)),]
17
18 #autoBuild svm model (or make our choices)
19 classifier = svm(formula = Gesture ~ .,
20                 data = alldata, gamma = 0.02, cost = 1000
21 )
22 write.svm(classifier,svm.file = "D:/Files/Desktop/newClass.txt")
23
```

ΕΙΚΟΝΑ 6.4.2.1: Το script που εκπαιδεύει το μοντέλο του SVM και δημιουργεί το αρχείο *newClass.txt*

Στη φόρμουλα που δημιούργησε το μοντέλο του SVM επιλέχθηκε ο πυρήνας RBF (Radial Basis Function) και ο τύπος SVM, C_SVC (Cost_SupportVectorClassification) και οι μεταβλητές gamma και cost (στη γραμμή 20) ορίστηκαν σε 0.02 και 1000 αντίστοιχα μέσα από μια διαδικασία trial & error αυξάνοντας και μειώνοντάς τες και αλλάζοντας πυρήνες και τύπους SVM μέχρι να βρεθεί ο κατάλληλος συνδυασμός που θα αποφέρει το μεγαλύτερο ποσοστό επιτυχίας στην αναγνώριση.

6.4.3 Κατηγοριοποίηση Χειρονομίας

Για την κατηγοριοποίηση της χειρονομίας κατά τη διαδικασία αναγνώρισης χρησιμοποιήθηκε η μέθοδος `svm_predict_values` που παρείχε η βιβλιοθήκη `libsvm` που ενσωματώθηκε στον κώδικα της εφαρμογής.

Με την εισαγωγή κάθε νέας χειρονομίας και κατά την επιλογή της αναγνώρισης της, με τη μέθοδο FFT, από το χρήστη η μέθοδος `svm_predict_values` χρησιμοποιεί την τακτική `one vs all` για την κατηγοριοποίηση σε μία από τις 5 κλάσεις, εφόσον οι διακριτές χειρονομίες είναι 5. Χρησιμοποιώντας τους Support Vectors που έχει αρχικοποιήσει σε πίνακες από το αρχείο **newClass.txt** συγκρίνει την απόσταση που έχουν τα σημεία της εισερχόμενης χειρονομίας με τους Support Vectors της κάθε κλάσης και υπολογίζει μια πιθανότητα ομοιότητας με την κάθε κλάση. Στη συνέχεια συγκρίνει τις πιθανότητες της κάθε κλάσης ανά ζεύγη με τον εξής τρόπο. π.χ. hh/hu, hh/hud, hh/hh2, hh/hu2, hu/hud, hu/hh2, hu/hu2, hud/hh2, hud/hu2, hh2/hu2

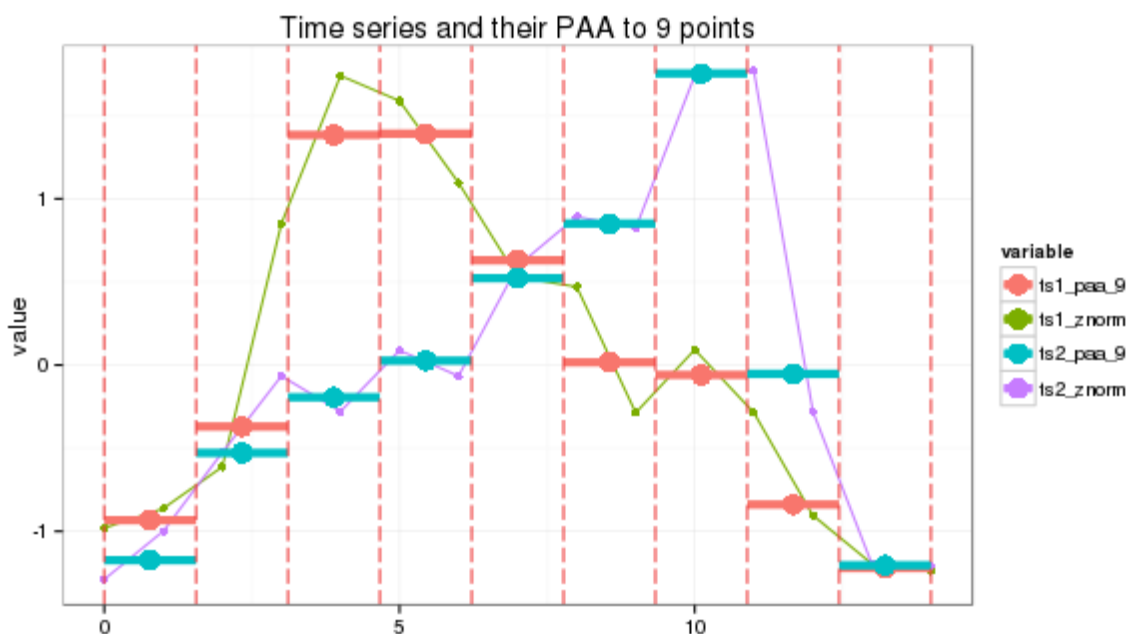
Όσες φορές μία κλάση έχει μεγαλύτερη πιθανότητα από την κλάση με την οποία συγκρίνεται παίρνει μια ψήφο ως πιο πιθανή για αναγνώριση και στο τέλος η κλάση στην οποία ανήκει η εισερχόμενη χειρονομία είναι αυτή με τις περισσότερες ψήφους.

6.5 Αλγόριθμος SAX

Ο αλγόριθμος SAX (Symbolic Aggregate approXimation) που προτάθηκε από την Jessica Lin κ.ά. [29] χρησιμοποιείται για την αναπαράσταση χρονοσειρών με σύμβολα. Η χρονοσειρά κβαντίζεται στον άξονα του χρόνου με τη βοήθεια της μεθόδου PAA (Piecewise Aggregate Approximation) ώστε να μειωθεί η διάσταση του χρόνου και στη συνέχεια κβαντίζεται στον άξονα των τιμών με σύμβολα ώστε να καταλήξει το σήμα να αναπαρίσταται ως συμβολοσειρά με διακριτές τιμές. Με αυτό τον τρόπο μπορούν να υπολογιστούν πιο εύκολα και πιο γρήγορα οι ομοιότητες και οι διαφορές ανάμεσα σε δύο χρονοσειρές, πράγμα που καθιστά τη χρήση του αλγόριθμου στην αναγνώριση χειρονομιών ως σημαντικό υποψήφιο για την επίτευξη μεγάλου ποσοστού ακρίβειας στην αναγνώριση σε πραγματικό χρόνο.

6.5.1 Επεξεργασία Δεδομένων

Για την επεξεργασία των δεδομένων στην επιλογή του αλγόριθμου SAX χρησιμοποιήθηκαν οι μέθοδοι PAA και SAX που παρέχει η βιβλιοθήκη SPMF που ενσωματώθηκε στον κώδικα της εφαρμογής. Η χρονοσειρά που αποτελείται από τις 90 τιμές από το επιταχυνσιόμετρο περνάει από κανονικοποίηση γύρω από το μηδέν και στη συνέχεια από μείωση διαστάσεων με τη μέθοδο PAA όπως φαίνεται στο παράδειγμα της εικόνας. [30]



ΕΙΚΟΝΑ 6.5.1.1: Μείωση διαστάσεων δύο χρονοσειρών σε 9 σημεία με τη μέθοδο PAA μετά από κανονικοποίηση των δεδομένων [30]

Στην περίπτωση του συγκεκριμένου συστήματος, οι 90 τιμές που αποτελούν τη χειρονομία, περιέχουν τα δεδομένα από 30 τιμές για κάθε άξονα X, Y, Z, από το επιταχυνσιόμετρο και μετά από επαναλαμβανόμενες δοκιμές επιλέχθηκε η μείωση της χρονοσειράς στα 30 σημεία που απέφερε τα καλύτερα αποτελέσματα. Αυτό ήταν αναμενόμενο καθώς το 30 είναι διαιρέτης του 90 και με αυτή τη μείωση επιτυγχάνουμε την ελάχιστη απώλεια πληροφορίας από τα δεδομένα καθώς η σημαντική πληροφορία των 30 τιμών του κάθε άξονα παραμένει σχεδόν ακέραια όταν μειωθεί στις 10 τιμές. Αυτό συμβαίνει διότι σύμφωνα με τον τρόπο λειτουργίας του αλγόριθμου PAA αν το πλήθος των σημείων που επιλεγεί για τη μείωση, δεν είναι διαιρέτης του πλήθους των δεδομένων, η μέση τιμή που θα υπολογιστεί για το σημείο από τα δεδομένα, στο τέλος των πρώτων 30 τιμών (άξονας X) και στην αρχή των επόμενων 30 τιμών (άξονας Y) θα περιέχει πληροφορία από τις τιμές και των δύο αξόνων. Το ίδιο πρόβλημα θα προέκυπτε και με τις τελευταίες τιμές του άξονα Y με τις αρχικές του άξονα Z, οπότε και αποφεύχθηκε με τον τρόπο που περιεγράφηκε.

Στη συνέχεια, η μειωμένη πλέον χρονοσειρά των 30 σημείων, μετατρέπεται σε μια συμβολοσειρά των 15 συμβόλων με τη βοήθεια του αλγορίθμου SAX ο οποίος ορίζει ένα σύμβολο σε κάθε σημείο της χρονοσειράς ανάλογα με το εύρος τιμών στο οποίο ανήκει. Το εύρος των τιμών υπολογίζεται με βάση το πλήθος των συμβόλων που θα αποτελείται η αναπαράσταση της χρονοσειράς. Η επιλογή του πλήθους των συμβόλων στα 15, έγινε επίσης μετά από επαναλαμβανόμενες δοκιμές σε συνδυασμό με τις δοκιμές για το πλήθος των σημείων του PAA μέχρι να βρεθεί η κατάλληλη ισορροπία που θα επιτύγχανε το μεγαλύτερο ποσοστό ακρίβειας στην αναγνώριση των χειρονομιών.

6.5.2 Προσπέλαση Δεδομένων

Κατά τη διάρκεια της δημιουργίας του data set, τα δεδομένα που καταγράφηκαν από την εκτέλεση της κάθε χειρονομίας αποθηκεύτηκαν στο αρχείο **saxInput.txt** καθώς η βιβλιοθήκη SPMF που χρησιμοποιήθηκε έδινε την επιλογή για μετασχηματισμό των χρονοσειρών από αρχείο τύπου .txt και την αποθήκευσή τους σε νέο αρχείο το οποίο ονομάστηκε **saxOutput.txt**. Το αρχείο που χρησιμοποιείται στην τελική έκδοση της εφαρμογής είναι το τελευταίο που περιέχει τις αναπαραστάσεις SAX και τα περιεχόμενά του αρχικοποιούνται σε έναν πίνακα κατά την έναρξη της εφαρμογής.

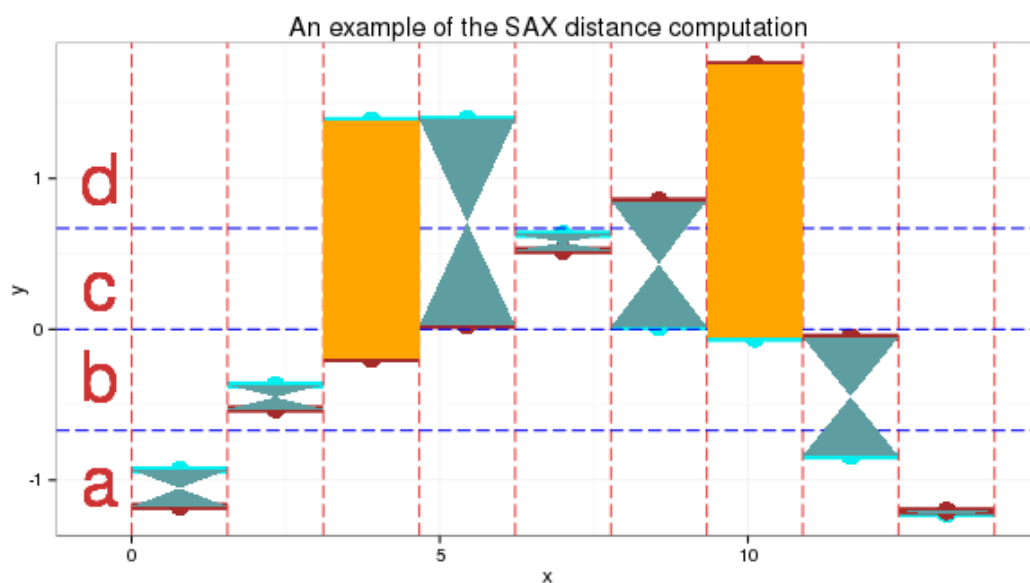
Για την καταγραφή νέων κινήσεων από τον εκάστοτε χρήστη που θέλει να εκπαιδεύσει το σύστημα ώστε να αναγνωρίζει τις χειρονομίες του με μεγαλύτερη ακρίβεια δημιουργήθηκαν δύο ακόμη αρχεία με την ίδια λειτουργικότητα όπως τα αρχεία που αναφέρθηκαν παραπάνω με όνομα **saxInputGlobal.txt** και **saxOutputGlobal.txt** αντίστοιχα. Τα τελευταία ενημερώνονται διαρκώς με την κάθε νέα καταγραφή χειρονομίας από το χρήστη και αρχικοποιούνται οι τιμές τους σε πίνακες στην έναρξη της εφαρμογής ώστε να μη χρειάζεται ο χρήστης να επαναλαμβάνει την καταγραφή των κινήσεων κάθε φορά.

6.5.3 Κατηγοριοποίηση Χειρονομίας

Η ταχύτητα και η ευκολία που προσφέρει η αναπαράσταση SAX για την αναγνώριση χειρονομιών βασίζεται στον τρόπο σύγκρισης μεταξύ δύο μετασχηματισμένων χρονοσειρών. Χρησιμοποιείται ένα παράδειγμα μικρότερης κλίμακας για την ευκολία της αναπαράστασης της διαδικασίας κατηγοριοποίησης.[31]

Κατ' επέκταση του παραδείγματος της εικόνας 6.5.1.1 οι αναπαραστάσεις των χρονοσειρών με τη μέθοδο SAX με 4 σύμβολα, σύμφωνα με την εικόνα που ακολουθεί, θα είναι:

- Αναπαράσταση Μπλε Χρονοσειράς: “abddccbaa”
- Αναπαράσταση Κόκκινης Χρονοσειράς: “abbccddba”



ΕΙΚΟΝΑ 6.5.3.1: Αναπαράσταση μειωμένων χρονοσειρών με τη μέθοδο SAX και ο υπολογισμός της απόστασής τους

Η απόσταση ανάμεσα σε 2 αναπαραστάσεις SAX με 4 σύμβολα υπολογίζεται με βάση τον πίνακα που ακολουθεί.

	a	b	c	d
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

ΕΙΚΟΝΑ 6.5.3.2: Πίνακας υπολογισμού αποστάσεων μεταξύ 2 αναπαραστάσεων SAX

Για να υπολογιστεί το άθροισμα της απόστασης μεταξύ των 2 χρονοσειρών, κάθε σύμβολο της μπλε αναπαράστασης συγκρίνεται με το αντίστοιχο σύμβολο της κόκκινης. Αν η απόσταση μεταξύ 2 συμβόλων είναι μηδέν ή ένα τότε δε λαμβάνεται υπόψιν στο άθροισμα. Αν τα σύμβολα έχουν απόσταση μεταξύ τους ίση με 2 τότε στο άθροισμα της απόστασης προστίθεται το 0.67 και αν απέχουν 3 σύμβολα ο αριθμός 1.34. Οι αριθμοί που προστίθενται στο άθροισμα της απόστασης είναι διαφορετικοί ανάλογα με το πλήθος των συμβόλων που επιλεχθεί για την αναπαράσταση των χρονοσειρών και προκύπτουν από ένα Matlab script που παρέχεται στην ιστοσελίδα των δημιουργών του αλγόριθμου: <https://www.cs.ucr.edu/~eamonn/SAX.htm>.

Για το συγκεκριμένο παράδειγμα η απόσταση ανάμεσα σε αυτές τις δύο χρονοσειρές είναι:

- Απόσταση SAX: $0 + 0 + 0.67 + 0 + 0 + 0 + 0.67 + 0 + 0 = 1.34$

Σύμφωνα με το παράδειγμα που προηγήθηκε, δημιουργήθηκε μια μέθοδος για τον υπολογισμό της απόστασης SAX κάθε νέας εισερχόμενης χειρονομίας είτε με τις χειρονομίες που περιέχει το καθολικό data set είτε με αυτές του προσωποκεντρικού που εκπαιδεύει ο χρήστης. Κατά την εκτέλεση της μεθόδου, το άθροισμα της απόστασης που υπολογίζεται με την κάθε σύγκριση αποθηκεύεται σε έναν πίνακα μεγέθους όσο και το πλήθος των αποθηκευμένων χειρονομιών. Στη συνέχεια γίνεται αναζήτηση σε αυτό τον πίνακα για την εύρεση του ελάχιστου αθροίσματος και επιστρέφεται η θέση στην οποία βρέθηκε. Από τη στιγμή που και οι δύο πίνακες έχουν το ίδιο πλήθος κελιών και την ίδια ταξινόμηση η θέση στην οποία ανιχνεύθηκε το μικρότερο άθροισμα αντιστοιχεί στη θέση του πίνακα που είναι αποθηκευμένες όλες οι χειρονομίες.

Με αυτόν τον τρόπο η εισερχόμενη χειρονομία κατηγοριοποιείται στην κλάση της αποθηκευμένης χειρονομίας που έχει τη μικρότερη απόσταση SAX από αυτήν.

ΚΕΦΑΛΑΙΟ 7: ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

7.1 Κατηγοριοποίηση Χειρονομιών

Για την αξιολόγηση του συστήματος όσον αφορά την απόδοσή του στην ακρίβεια αναγνώρισης χειρονομιών από ανεξάρτητους χρήστες συντάχθηκε ένα data set με 250 εγγραφές χειρονομιών τις οποίες εκτέλεσαν 5 άτομα διαφορετικού φύλου, ηλικίας, ύψους και αναλογιών, σε διαφορετικά χρονικά παράθυρα, με διαφορετική ταχύτητα και σε κάποιες επαναλήψεις με μια ελαφρά κλίση στον καρπό. Η καταγραφή έγινε στο χρονικό διάστημα της μίας εβδομάδας.

- 5 άτομα * 5 χειρονομίες * 10 επαναλήψεις ανά χειρονομία → 250 εγγραφές

Επιλέχθηκε αυτή η προσέγγιση εφόσον ένας από τους κύριους στόχους στην ανάπτυξη αυτού του συστήματος ήταν η χρήση ενός μικρότερου δείγματος, σε σχέση με τους μελέτες που είχαν προηγηθεί, με την προσπάθεια να επιτευχθεί το ίδιο ή και μεγαλύτερο ποσοστό ακρίβειας στην αναγνώριση από ανεξάρτητο χρήστη. Με τη συμμετοχή διαφορετικών ατόμων στο δείγμα με τα χαρακτηριστικά και τους τρόπους εκτέλεσης που αναφέρθηκαν καλύφθηκε ένα μεγάλο εύρος πιθανών τρόπων εκτέλεσης που αξιολογήθηκε από 8 άτομα εκ των οποίων τα 5 ήταν αυτά που συμμετείχαν στη σύνταξη του δείγματος.

Στη λειτουργία κατηγοριοποίησης των χειρονομιών με τον αλγόριθμο SAX ο χρήστης έχει την επιλογή να αποθηκεύσει τις χειρονομίες που θα εκτελεί σε ένα προσωπικό αρχείο και να γίνεται αναγνώριση με βάση το αρχείο αυτό. Στην αξιολόγηση αυτής της λειτουργίας συμμετείχαν 8 άτομα που εκπαιδύσαν το σύστημα ο καθένας ξεχωριστά με 2 τρόπους.

- Μία εκτέλεση της κάθε χειρονομίας (συνολικό πλήθος χειρονομιών στο δείγμα: 5)
- Πέντε εκτελέσεις της κάθε χειρονομίας (συνολικό πλήθος χειρονομιών στο δείγμα: 25)

Δεν έγινε αξιολόγηση με μεγαλύτερο δείγμα εκπαίδευσης καθώς σε ένα πραγματικό σενάριο χρήσης της εφαρμογής θα ήταν εξαιρετικά χρονοβόρο να εκτελεί ο χρήστης περισσότερες από 5 φορές την κάθε χειρονομία ενώ θα είχε ελάχιστη αύξηση στο ποσοστό επιτυχίας αναγνώρισης σύμφωνα με τα αποτελέσματα που παρατηρήθηκαν.

Στις παρακάτω υποενότητες ακολουθούν αναλυτικά τα αποτελέσματα της αξιολόγησης της απόδοσης των αλγορίθμων στην αναγνώριση των χειρονομιών.

7.1.2 Κατηγοριοποίηση με τον FFT

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, για την κατηγοριοποίηση των χειρονομιών με τον αλγόριθμο FFT χρησιμοποιήθηκε μία μηχανή υποστήριξης διανυσμάτων που είχε εκπαιδευτεί με το καθολικό δείγμα χειρονομιών που περιέχει 250 εγγραφές.

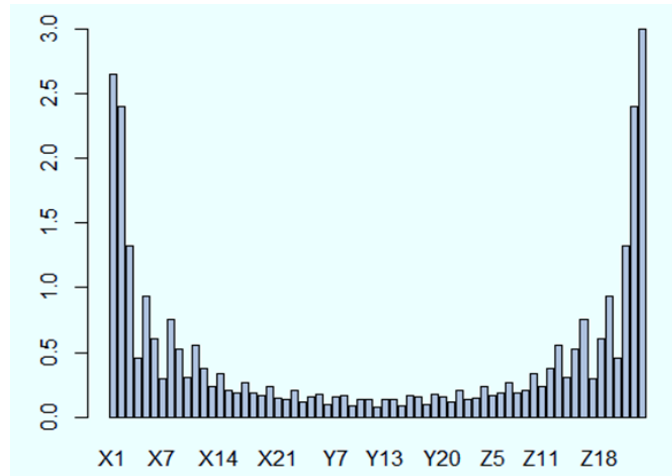
Τα αποτελέσματα της αξιολόγησης του αλγόριθμου FFT με σύνολο 250 δοκιμές από τα 5 άτομα που συμμετείχαν στο δείγμα φαίνονται στον παρακάτω πίνακα.

FFT	hh	hu	hud	hh2	hu2
hh	50	0	0	0	0
hu	0	50	0	0	0
hud	0	2	48	0	0
hh2	5	0	0	45	0
hu2	0	1	0	0	49

ΕΙΚΟΝΑ 7.1.2.1: Συγκεντρωτικός πίνακας αποτελεσμάτων αναγνώρισης χειρονομιών με τον αλγόριθμο FFT από άτομα που συμμετείχαν στο δείγμα

Μετά από δοκιμές και ρυθμίσεις των μεταβλητών του SVM, το ποσοστό ακρίβειας που επιτεύχθηκε στην αναγνώριση χειρονομιών από τα άτομα που συμμετείχαν στην σύνταξη του δείγματος είναι **96.8%** ενώ σε 2 από τους 5 χρήστες το ποσοστό επιτυχίας ήταν **100%**.

Παρατηρούμε πως, οι ψευδώς αληθείς κατηγοριοποιήσεις που έγιναν, ανήκουν στις σύνθετες χειρονομίες οι οποίες περιέχουν εν μέρει την ίδια πληροφορία με τις απλές. Συγκεκριμένα, το πρώτο μισό της εκτέλεσης της χειρονομίας hud περιέχει το πρώτο μισό της hu, ενώ οι hh2 και hu2 είναι η διπλή εκτέλεση των χειρονομιών hh και hu αντίστοιχα. Η μεγαλύτερη αστοχία στην αναγνώριση εμφανίστηκε στον αλγόριθμο hh2 καθώς με έναν σύντομο έλεγχο στα ωμά δεδομένα του επιταχυνσιόμετρου, που ανήκουν στις χειρονομίες hh και hh2, δεν παρατηρούνται σημαντικές διαφορές. Κατ' επέκταση το ίδιο συμβαίνει και στα δεδομένα που χρησιμοποιούνται στην εκπαίδευση του SVM μετά το μετασχηματισμό FFT όπως φαίνεται στη γραφική παράσταση που ακολουθεί.



ΕΙΚΟΝΑ 7.1.2.2: Τα μέτρα των μιγαδικών αριθμών που προκύπτουν από το μετασχηματισμό Fourier των δεδομένων του επιταχυνσιόμετρου

Πιο συγκεκριμένα, παρατηρούμε πως οι μετασχηματισμένες τιμές του άξονα του Y έχουν πολύ μικρές διαφορές μεταξύ τους καθώς ανήκουν όλες σχεδόν στο ίδιο εύρος τιμών και κατά την εκτέλεση των χειρονομιών hh και hh2 το χέρι του χρήστη κινείται κυρίως κατά μήκος του άξονα των Y. Συνεπώς το SVM έχει δώσει μεγαλύτερο βάρος στα δεδομένα του άξονα του Y για να μπορέσει να κατηγοριοποιήσει αυτές τις δύο χειρονομίες και λόγω της ομοιότητας των δεδομένων τους εμφανίζει ψευδώς αληθή αποτελέσματα περισσότερες φορές σε σχέση με τις άλλες χειρονομίες.

Τα αποτελέσματα της αξιολόγησης του αλγόριθμου FFT με σύνολο 150 δοκιμές από 3 άτομα που δε συμμετείχαν στο δείγμα φαίνονται στον παρακάτω πίνακα.

FFT	hh	hu	hud	hh2	hu2
hh	30	0	0	0	0
hu	0	30	0	0	0
hud	0	1	29	0	0
hh2	6	0	0	24	0
hu2	0	2	0	0	28

ΕΙΚΟΝΑ 7.1.2.3: Συγκεντρωτικός πίνακας αποτελεσμάτων αναγνώρισης χειρονομιών με τον αλγόριθμο FFT από άτομα που δε συμμετείχαν στο δείγμα

Το ποσοστό ακρίβειας που επιτεύχθηκε σε αυτή την αξιολόγηση είναι **94%**, πράγμα που σημαίνει πως το δείγμα που συντάχθηκε είναι αρκετά αναλυτικό για να αναγνωρίζει με τόση επιτυχία και ανεξάρτητους χρήστες. Παρατηρείται επίσης το ίδιο μοτίβο στις ψευδώς αληθείς κατηγοριοποιήσεις που επιβεβαιώνει το συμπέρασμα που αναλύθηκε παραπάνω.

7.1.3 Κατηγοριοποίηση με τον SAX

Η ανάλυση της αξιολόγησης της απόδοσης του αλγόριθμου SAX χωρίζεται σε δύο υποενότητες. Την αξιολόγηση της αναγνώρισης χειρονομιών με το καθολικό δείγμα που προαναφέρθηκε και την αξιολόγηση της επιτυχίας της αναγνώρισης με το προσωποκεντρικό δείγμα που συνέταξε κάθε ένας από τα 8 άτομα ξεχωριστά.

7.1.3.1 Κατηγοριοποίηση με Καθολικό Training Set

Τα αποτελέσματα της αξιολόγησης του αλγόριθμου SAX με σύνολο 250 δοκιμές από τα 5 άτομα που συμμετείχαν στο δείγμα φαίνονται στον παρακάτω πίνακα.

SAX	hh	hu	hud	hh2	hu2
hh	50	0	0	0	0
hu	0	50	0	0	0
hud	0	0	50	0	0
hh2	3	0	0	47	0
hu2	0	0	0	0	50

ΕΙΚΟΝΑ 7.1.3.1.1: Συγκεντρωτικός πίνακας αποτελεσμάτων αναγνώρισης χειρονομιών με τον αλγόριθμο SAX από άτομα που συμμετείχαν στο δείγμα

Μετά από δοκιμές που έγιναν κατά τη διάρκεια ανάπτυξης του συστήματος για τον ορισμό του πλήθους των σημείων PAA και των συμβόλων για την αναπαράσταση SAX, το ποσοστό επιτυχίας στην αναγνώριση των χειρονομιών έφτασε το **98.8%** με τις 3 ψευδώς αληθείς κατηγοριοποιήσεις να ανήκουν σε ένα άτομο. Η αξιολόγηση από τα υπόλοιπα 4 άτομα είχε ποσοστό επιτυχίας **100%**.

Όσον αφορά τις ψευδώς αληθείς κατηγοριοποιήσεις περιορίζονται στη χειρονομία hh2 την οποία ο αλγόριθμος αναγνώρισε ως την hh επιβεβαιώνοντας ακόμη μια φορά την ομοιότητα των δεδομένων αυτών των 2 χειρονομιών.

Το ίδιο ικανοποιητικά αποτελέσματα παρουσίασε και η αξιολόγηση του αλγορίθμου από 150 δοκιμές τριών ατόμων που δε συμμετείχαν στο δείγμα, όπως φαίνονται στον πίνακα που ακολουθεί.

SAX	hh	hu	hud	hh2	hu2
hh	30	0	0	0	0
hu	0	30	0	0	0
hud	0	0	30	0	0
hh2	5	0	0	25	0
hu2	0	1	0	0	29

ΕΙΚΟΝΑ 7.1.3.1.2: Συγκεντρωτικός πίνακας αποτελεσμάτων αναγνώρισης χειρονομιών με τον αλγόριθμο SAX από άτομα που δε συμμετείχαν στο δείγμα

Το ποσοστό επιτυχίας του αλγορίθμου SAX στην αναγνώριση χειρονομιών από ανεξάρτητους χρήστες έφτασε το **96%** για κάθε έναν από τους 3 χρήστες. Αυτό επαληθεύει το πόσο αναλυτικό είναι το δείγμα που συντάχθηκε εφόσον σε συνδυασμό με την ακρίβεια που προσφέρει ο αλγόριθμος κατηγοριοποιήθηκαν ως ψευδώς αληθείς μόνο 6 στις 150 δοκιμές. Η πλειοψηφία των τελευταίων ανήκει στη χειρονομία hh2 την οποία ο αλγόριθμος αναγνώρισε ως την hh, με μόνο μία εκτέλεση της hu2 να αναγνωρίζεται λανθασμένα ως hu. Αυτό παρατηρήθηκε και σε όλες τις υπόλοιπες δοκιμές κατά την αξιολόγηση του συστήματος εφόσον τα δεδομένα των δύο αυτών χειρονομιών μοιάζουν αρκετά μεταξύ τους και η χειρονομία hu2 είναι η διπλή εκτέλεση της hu.

7.1.3.2 Κατηγοριοποίηση με Προσωποκεντρικό Training Set

Για την αξιολόγηση της αναγνώρισης χειρονομιών με το προσωποκεντρικό δείγμα, ο κάθε χρήστης δημιουργούσε ένα νέο αρχείο που περιείχε μόνο τις δικές του εκτελέσεις από την κάθε χειρονομία. Αυτό έγινε σε δύο στάδια για κάθε ένα από τα 8 άτομα που συμμετείχαν στην αξιολόγηση. Για να είναι το κείμενο πιο ευανάγνωστο αναφέρονται μόνο τα ποσοστά επιτυχίας των χρηστών και ακολουθεί ένας συγκεντρωτικός πίνακας με όλα τα αποτελέσματα.

Στο πρώτο στάδιο ο κάθε χρήστης κατέγραψε την κάθε χειρονομία μόνο μία φορά και ακολούθησαν 5 εκτελέσεις της κάθε χειρονομίας για να υπολογιστεί το ποσοστό ακρίβειας στην αναγνώριση με μόνο ένα δείγμα. Σε 5 από τους 8 χρήστες ο αλγόριθμος σημείωσε ποσοστό επιτυχίας **100%** ενώ στους άλλους 2 είχε επιτυχία **96%** και σε έναν σημείωσε το ελάχιστο ποσοστό επιτυχίας **92%**. Τα αποτελέσματα φαίνονται αναλυτικά στον ακόλουθο πίνακα.

SAX	hh	hu	hud	hh2	hu2
hh	40	0	0	0	0
hu	0	40	0	0	0
hud	0	0	40	0	0
hh2	3	0	0	37	0
hu2	0	1	0	0	39

ΕΙΚΟΝΑ 7.1.3.2.1: Συγκεντρωτικός πίνακας αποτελεσμάτων αναγνώρισης χειρονομιών με τον αλγόριθμο SAX από άτομα που συνέταξαν ο καθένας το δικό του δείγμα με μία καταγραφή

Συνεπώς το συνολικό ποσοστό επιτυχίας του αλγορίθμου SAX στην αναγνώριση χειρονομιών με μία μόνο καταγραφή της κάθε χειρονομίας φτάνει το **98%** και οι ψευδώς αληθείς κατηγοριοποιήσεις βρίσκονται, σύμφωνα με το μοτίβο, στις χειρονομίες hh2 και hu2.

Στο δεύτερο στάδιο έπρεπε να μελετηθεί το πόσο μπορεί να βελτιωθεί το ποσοστό ακρίβειας στην αναγνώριση με την προσθήκη νέων καταγραφών του κάθε χρήστη στο προσωποκεντρικό του αρχείο. Για το λόγο αυτό ο κάθε χρήστης κατέγραψε άλλες 4 εκτελέσεις για την κάθε χειρονομία στο αρχείο του και ακολούθησαν άλλες 5 δοκιμές για την αναγνώριση της κάθε χειρονομίας. Στο τέλος αυτής της διαδικασίας το ποσοστό επιτυχίας της αναγνώρισης ήταν **100%** για 7 από τα 8 άτομα και **96%** για τον τελευταίο χρήστη. Τα αποτελέσματα φαίνονται αναλυτικά στον παρακάτω πίνακα.

SAX	hh	hu	hud	hh2	hu2
hh	40	0	0	0	0
hu	0	40	0	0	0
hud	0	0	40	0	0
hh2	0	0	0	40	0
hu2	0	1	0	0	39

ΕΙΚΟΝΑ 7.1.3.2.1: Συγκεντρωτικός πίνακας αποτελεσμάτων αναγνώρισης χειρονομιών με τον αλγόριθμο SAX από άτομα που συνέταξαν ο καθένας το δικό του δείγμα με 5 καταγραφές

Συνεπώς το συνολικό ποσοστό επιτυχίας του αλγορίθμου SAX στην αναγνώριση χειρονομιών με μία μόνο καταγραφή της κάθε χειρονομίας φτάνει το **99.5%** και η μία ψευδώς αληθής κατηγοριοποίηση ανήκει στην χειρονομία hu2 την οποία αναγνώρισε ως την hu.

Το ποσοστό ακρίβειας του αλγορίθμου SAX στην αληθώς αληθή κατηγοριοποίηση των χειρονομιών ήταν αρκετά υψηλό με την καταγραφή μίας μόνο χειρονομίας και έφτασε σχεδόν το 100% με συνολικά 5 καταγραφές. Σε αυτό το σημείο παρατηρείται πως ο αλγόριθμος μπορεί να λειτουργήσει καλύτερα με μικρότερο δείγμα που έχει εκπαιδευτεί από τον κάθε χρήστη με τις προσωπικές του χειρονομίες, παρά με ένα μεγάλο δείγμα που περιέχει πολλές χειρονομίες διαφορετικών ατόμων καθώς το ποσοστό επιτυχίας αναγνώρισης με το προσωποκεντρικό δείγμα με 5 καταγραφές της κάθε χειρονομίας είναι μεγαλύτερο (99.5%) από το αντίστοιχο της αναγνώρισης των 5 ατόμων που συμμετείχαν στο καθολικό δείγμα (98.8%).

7.2 Συγκεντρωτικά Αποτελέσματα

Τα αποτελέσματα φαίνονται συγκεντρωτικά στους πίνακες 7.2.1, 7.2.2 και 7.2.3. Από την ανάλυση των αποτελεσμάτων προκύπτουν τα ακόλουθα συμπεράσματα.

Τα «ωμά» δεδομένα από το επιταχυνσιόμετρο είναι αρκετά ακριβή ώστε να επιτρέπουν τη σύνταξη ενός δείγματος χειρονομιών το οποίο μπορεί να αποφέρει μεγάλο ποσοστό επιτυχίας στην αναγνώριση χειρονομιών και με τους δύο αλγόριθμους που υλοποιήθηκαν.

Το καθολικό δείγμα των 250 εγγραφών που δημιουργήθηκε συνέβαλε αρκετά στην ακρίβεια κατά την αξιολόγησή του στην αναγνώριση χειρονομιών από χρήστες που συμμετείχαν στη σύνταξή του και από ανεξάρτητους χρήστες καθώς με τον τρόπο που έγινε η καταγραφή των χειρονομιών έγινε προσπάθεια για να καλυφθεί η μεγαλύτερη δυνατή ποικιλία στις αναλογίες των χρηστών και στον τρόπο εκτέλεσης των χειρονομιών.

Recognition Algorithm	True	False	%	%max	%min
FFT User Dependent	242	8	96.8	100	92
FFT User Independent	141	9	94	96	92
SAX User Dependent	247	3	98.8	100	94
SAX User Independent	144	6	96	96	96

ΕΙΚΟΝΑ 7.2.1: Συγκεντρωτικός πίνακας ποσοστών επιτυχίας αναγνώρισης χειρονομιών των 2 αλγορίθμων με χρήστες που συμμετείχαν στο δείγμα και με ανεξάρτητους χρήστες

Στην κατηγοριοποίηση των χειρονομιών με το καθολικό δείγμα ο αλγόριθμος SAX έδειξε μεγαλύτερο ποσοστό επιτυχίας από τον FFT στην περίπτωση αξιολόγησης από τους χρήστες που συνέταξαν το δείγμα αλλά και στην αξιολόγηση από ανεξάρτητους χρήστες πράγμα που ήταν αναμενόμενο καθώς στη σχετική βιβλιογραφία ο αλγόριθμος SAX έχει μεγαλύτερο ποσοστό επιτυχίας από τους περισσότερους αλγόριθμους ειδικά στην αναγνώριση χειρονομιών από ανεξάρτητους χρήστες.

SAX Method	True	False	%	%max	%min
Global Data Set	242	8	96.8	100	92
Personal Data Set One Recording	196	4	98	100	92
Personal Data Set Five Recordings	199	1	99.5	100	96

ΕΙΚΟΝΑ 7.2.2: Συγκεντρωτικός πίνακας ποσοστών επιτυχίας αναγνώρισης του αλγόριθμου SAX με το καθολικό δείγμα και με προσωποκεντρικό δείγμα με μία και πέντε εγγραφές

Επιπροσθέτως, ο αλγόριθμος SAX επιτυγχάνει μεγαλύτερο ποσοστό ακρίβειας στην αναγνώριση όταν χρησιμοποιείται μικρότερο δείγμα για τη σύγκριση της νέας εισερχόμενης χειρονομίας και πιο συγκεκριμένα όταν χρησιμοποιείται προσωποκεντρικό δείγμα που έχει εκπαιδεύσει ο ίδιος εκτελώντας την κάθε χειρονομία από μία έως πέντε φορές.

Gesture	hh	hu	hud	hh2	hu2
hh	240	0	0	0	0
hu	0	240	0	0	0
hud	0	3	237	0	0
hh2	22	0	0	218	0
hu2	0	6	0	0	234

ΕΙΚΟΝΑ 7.2.3: Συγκεντρωτικός πίνακας του συνόλου των χειρονομιών που εκτελέστηκαν κατά την αξιολόγηση

Τέλος, όσον αφορά τις διακριτές χειρονομίες και οι 2 αλγόριθμοι αναγνώρισαν επιτυχώς σε κάθε δοκιμή τις απλές χειρονομίες hh και hu με ελάχιστες ψευδώς αληθείς κατηγοριοποιήσεις της χειρονομίας hud ως hu. Παρατηρήθηκε επίσης ένα μοτίβο για τις ψευδώς αληθείς κατηγοριοποιήσεις των σύνθετων χειρονομιών hh2 και hu2 οι οποίες αποτελούνται από τη διπλή εκτέλεση των απλών χειρονομιών hh και hu αντίστοιχα, πράγμα που συνέβαλε στη λάθος αναγνώριση, με τη χειρονομία hh2 να αναγνωρίζεται εσφαλμένα, ως hh 22 φορές, περισσότερο από όλες τις υπόλοιπες χειρονομίες. Αυτό οφείλεται στην ομοιότητα που έχουν τα δεδομένα αυτών των δύο χειρονομιών στους 3 άξονες του επιταχυνσιόμετρου και κυρίως στον άξονα Y που κινείται το χέρι του χρήστη κατά την εκτέλεση τους.

ΚΕΦΑΛΑΙΟ 8: ΕΠΙΛΟΓΟΣ

8.1 Συμπεράσματα

Ολοκληρώνοντας τη διπλωματική εργασία, συμπεραίνεται πως το κάθε βήμα σχεδίασης και ανάπτυξης του συστήματος είναι εξίσου σημαντικό με τα υπόλοιπα ώστε το τελικό αποτέλεσμα να μην αποκλίνει από τον αρχικό στόχο.

Το αρχικό βήμα, ήταν η μελέτη σχετικών ερευνών στη βιβλιογραφία με θέμα την αναγνώριση χειρονομιών και των διαθέσιμων αλγορίθμων για την αναγνώρισή τους όπως και η κατανόηση αλγορίθμων επεξεργασίας σημάτων, με σκοπό να δημιουργηθεί το κατάλληλο υπόβαθρο γνώσεων μέσω της υφιστάμενης κατάστασης. Αφού ολοκληρώθηκε η έρευνα για τις σχετικές μελέτες, αξιολογήθηκε ο τρόπος ανάπτυξης του κάθε συστήματος που υλοποιήθηκε και σχεδιάστηκε το προτεινόμενο σύστημα περιλαμβάνοντας τα πλεονεκτήματα και αποφεύγοντας τα μειονεκτήματα της κάθε μελέτης, σε μία προσπάθεια βελτίωσης των αποτελεσμάτων.

Εφόσον εντοπίστηκαν οι αδυναμίες και οι δυνατότητες των προσεγγίσεων που μελετήθηκαν, ακολούθησε η καταγραφή των απαιτήσεων και οι λειτουργίες που θα προσφέρει το προτεινόμενο σύστημα, δηλαδή τις προϋποθέσεις που απαιτεί το σύστημα για τη σωστή λειτουργία του από πλευράς hardware και software και οι δυνατότητες που θα έχει.

Για την υλοποίηση του συστήματος χρησιμοποιήθηκε μία συσκευή κινητού υπολογισμού και μια συσκευή φορητής υπολογιστικής ικανότητας που για τη μεταξύ τους επικοινωνία χρησιμοποιήθηκε η τεχνολογία Bluetooth. Το σύστημα είναι σε θέση να αναγνωρίσει 5 διακριτές χειρονομίες από διαφορετικούς χρήστες. Για την αναγνώριση των χειρονομιών από τα δεδομένα του επιταχυνσιόμετρου που στέλνει το έξυπνο ρολόι στο έξυπνο κινητό έγινε χρήση των αλγορίθμων FFT και SAX με τον δεύτερο να έχει τη μεγαλύτερη απόδοση με ποσοστό επιτυχίας 98.8% και 96% στην αναγνώριση από χρήστες που συμμετείχαν στην εκπαίδευση του δείγματος και σε ανεξάρτητους χρήστες αντίστοιχα. Στην περίπτωση που ο χρήστης επιλέξει να συντάξει το προσωποκεντρικό του δείγμα, εκτελώντας 5 φορές την κάθε χειρονομία, η ακρίβεια ανεβαίνει στο 99.5%.

Εν κατακλείδι, μέσω της εκπόνησης της παρούσας διπλωματικής εργασίας υλοποιήθηκε ένα σύστημα για εφαρμογές διάχυτου υπολογισμού στην προσπάθεια του να απεξηρτηθεί ο χρήστης από το τρίπτυχο ποντίκι, πληκτρολόγιο, οθόνη που απαιτείται για την επικοινωνία ανθρώπου υπολογιστή, χρησιμοποιώντας πλέον κινήσεις του σώματός του για την αλληλεπίδραση με το εκάστοτε σύστημα. Το σύστημα που υλοποιήθηκε ικανοποιεί τον αρχικό στόχο που τέθηκε, δηλαδή την αξιόπιστη αναγνώριση των 5 διακριτών χειρονομιών με μικρότερο δείγμα αλλά και τη δυνατότητα εκπαίδευσης του συστήματος από το χρήστη για την επίτευξη της μέγιστης δυνατής ακρίβειας.

8.2 Μελλοντικές Βελτιώσεις

Μολονότι το σύστημα δεν έχει σημαντικές αδυναμίες και πετυχαίνει τον αρχικό του στόχο, σε μία προσπάθεια να αποφευχθούν οι αδυναμίες των ερευνών που μελετήθηκαν προέκυψαν οι ακόλουθες αδυναμίες:

- Για την αποφυγή υλοποίησης μεθόδων όπως φίλτρα για την απόσπαση της σημαντικής πληροφορίας από ένα σήμα που περιέχει θόρυβο ή είναι συνεχές, ο χρήστης πρέπει να πιέζει ένα κουμπί στην οθόνη της συσκευής κινητού υπολογισμού πριν την έναρξη της εκτέλεσης της κάθε χειρονομίας, το οποίο ξεκινά την καταγραφή των δεδομένων. Για την αντιμετώπιση αυτού του προβλήματος μπορεί να υλοποιηθεί μια μέθοδος στη συσκευή φορητής υπολογιστικής ικανότητας η οποία θα ανιχνεύει την κλίση του καρπού ώστε ο χρήστης να ξεκινά τη διαδικασία της αποστολής των δεδομένων με τη χειρονομία wrist tilt και στη συνέχεια να εκτελεί τη χειρονομία.
- Με την υλοποίηση της μεθόδου που προαναφέρθηκε μπορεί να αντιμετωπιστεί και το πρόβλημα της υπερβολικής κατανάλωσης μπαταρίας της συσκευής φορητής υπολογιστικής ικανότητας καθώς, στην παρούσα μορφή του συστήματος, η εφαρμογή της τελευταίας στέλνει διαρκώς δεδομένα στο μονοπάτι του Bluetooth με αποτέλεσμα το ποσοστό πτώσης της στάθμης της μπαταρίας να ανέρχεται στο 20% ανά ώρα λειτουργίας.
- Όσον αφορά τον τρόπο κατηγοριοποίησης των χειρονομιών, οι δύο αλγόριθμοι που υλοποιήθηκαν δε διαθέτουν κάποια τεχνική για την αποφυγή ψευδώς αληθών αναγνωρίσεων με αποτέλεσμα να αναγνωρίζεται πάντα μια από τις 5 διακριτές χειρονομίες ακόμη και αν ο χρήστης δεν έχει εκτελέσει σωστά κάποια από αυτές. Για αντιμετωπιστεί το πρόβλημα αυτό, θα πρέπει να γίνει περεταίρω ανάλυση του συστήματος και να τροποποιηθεί η υλοποίηση των αλγορίθμων ώστε να ανιχνεύονται αληθώς μόνο οι 5 διακριτές χειρονομίες που προσφέρει το σύστημα.
- Ένα ακόμη μειονέκτημα που ανιχνεύθηκε είναι πως ο τρόπος υλοποίησης του συστήματος απαιτούσε από το χρήστη να πιέσει ένα ακόμη κουμπί στην οθόνη της συσκευής κινητού υπολογισμού για να επιλέξει τον αλγόριθμο με τον οποίο επιθυμεί να αναγνωριστεί η χειρονομία που έχει εκτελέσει. Εφόσον, σύμφωνα με τα αποτελέσματα, ο αλγόριθμος που προσφέρει τη μεγαλύτερη ακρίβεια είναι ο SAX, θα μπορούσε να υλοποιηθεί μία νέα εφαρμογή που θα χρησιμοποιεί μόνο αυτόν τον αλγόριθμο ο οποίος θα εκτελείται αυτόματα μετά το πέρας της εκτέλεσης της χειρονομίας χωρίς να χρειάζεται επιπλέον αλληλεπίδραση από το χρήστη.

Λαμβάνοντας υπόψη τις αδυναμίες του συστήματος που προαναφέρθηκαν, ακολουθούν κάποιες προτάσεις που θα βελτιώσουν μελλοντικά το σύστημα:

- Αύξηση του αριθμού των διακριτών χειρονομιών με σκοπό την προσθήκη νέων λειτουργιών για ανάθεση σε κάθε χειρονομία χωρίς τη μείωση της ακρίβειας.
- Αύξηση του καθολικού δείγματος χειρονομιών με την προϋπόθεση να μην ξεπεράσει τις 500 εγγραφές, σε μία προσπάθεια να συμπεριληφθούν ακόμα περισσότερες εκτελέσεις χειρονομιών από άτομα διαφορετικών αναλογιών και με διαφορετικούς τρόπους όπως περιεγράφηκε στην εργασία.
- Ανάθεση κάποιας λειτουργίας σε κάθε χειρονομία σε μια πιθανή ενσωμάτωση του συστήματος σε εγκατάσταση βοηθητικής διαβίωσης, ένα έξυπνο σπίτι ή για την ανέπαφη αλληλεπίδραση με την ίδια τη συσκευή κινητού υπολογισμού.

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ

- [1] Mark Weiser (1991). The Computer for the 21st Century. Scientific American Ubicomp Paper.
- [2] Νικόλαος Κατερτζής (2018). Υπολογιστική συνείδηση και διάχυτος υπολογισμός. Μελέτη εφικτότητας ήρεμου και ευφυσούς υπολογισμού μέσω της τροποποίησης του προγράμματος ELIZA. Ελληνικό Ανοικτό Πανεπιστήμιο.
- [3] “Smartwatch”. Διαθεσιμότητα: <https://en.wikipedia.org/wiki/Smartwatch>
- [4] “Huawei Watch 2”. Διαθεσιμότητα: <https://consumer.huawei.com/gr/wearables/watch2/specs/>
- [5] Valeria Villani, Lorenzo Sabattini, Giuseppe Riggio, Alessio Levratti, Cristian Secchi, Cesare Fantuzzi (2017) Interacting With a Mobile Robot with a Natural Infrastructure-Less Interface. Διαθεσιμότητα: <https://arxiv.org/pdf/1704.00905.pdf>
- [6] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang ‘Anthony’ Chen, Xiaojun Bi, Xing-Dong Yang (2018) WrisText: One-handed Text Entry on Smartwatch using Wrist Gestures. Διαθεσιμότητα: <https://www.cs.dartmouth.edu/~xingdong/papers/WrisText.pdf>
- [7] Tran Huy Vu, Archan Misra, Quentin Roy, Kenny Tsu Wei Choo, Youngki Lee (2018) Smartwatch-based early gesture detection & trajectory tracking for interactive gesture-driven applications. Διαθεσιμότητα: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=5256&context=sis_research
- [8] Michael Xie, David Pan – University of Stanford – Accelerometer Gesture Recognition.
- [9] Zhenyu He – Computer Center, Jinan University, Guangzhou, China – Accelerometer Based Gesture Recognition Using Fusion Features and SVM.
- [10] Jungsoo Kim, Jiasheng He, Kent Lyons, Thad Starner - College of Computing and GVU Center Georgia Institute of Technology Atlanta, GA 30332-0280 USA – The Gesture Watch: A Wireless Contact-Free Gesture Based Wrist Interface.
- [11] Jiayang Liu, Zhen Wang and Lin Zhong - Department. Of Electrical Computer Engineering Rice University, Houston TX 77005, Jehan Wickramasuriya and Venu Vasudevan - Pervasive Platforms & Architectures Lab Applications & Software Research Center, Motorola Labs – uWave: Accelerometer-Based Personalized Gesture Recognition and its Applications
- [12] Jianhui Wu, Gang Pan, Daqing Zhang, Guande Qi, Shijian Li – Department of Computer Science, Zhejiang University, Hangzhou, 310027, China, Handicom Lab, Institut TELECOM SudParis, France – Gesture Recognition with a 3-D Accelerometer
- [13] Xu Zhang Chen, Associate Member, IEEE, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang – A Framework for Hand Gestures Recognition Based on Accelerometer and EMG Sensors

- [14] Thomas Schlömer, Benjamin Poppinga, Niels Henze, Susanne Boll – University of Oldenburg, OFFIS Institute for Information Technology – Gesture Recognition with a Wii Controller
- [15] Ahmad Akl and Shahrokh Valaee – Department of Electrical and Computer Engineering, University of Toronto – Accelerometer-Based Gesture Recognition via Dynamic-Time Warping, Affinity Propagation, & Compressive Sensing
- [16] “Android OS” – Διαθεσιμότητα: <https://el.wikipedia.org/wiki/Android>
- [17] “Wear OS” – Διαθεσιμότητα: https://en.wikipedia.org/wiki/Wear_OS
- [18] “Διάγραμμα Ροής Δεδομένων” – Διαθεσιμότητα: https://el.wikipedia.org/wiki/Διάγραμμα_ροής_δεδομένων
- [19] “Ενοποιημένη Γλώσσα Σχεδίασης Προτύπων (UML)” – Διαθεσιμότητα: https://el.wikipedia.org/wiki/Ενοποιημένη_Γλώσσα_Σχεδίασης_Προτύπων
- [20] Mobile Operating System Market Share Worldwide, Sept 2018 – Sept 2019
Διαθεσιμότητα: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [21] Huawei Honor 9 Lite Specifications - https://www.gsmarena.com/honor_9_lite-8962.php
- [22] Smartwatch market share of overall wearable market worldwide in 2018 and 2022, by operating system – Διαθεσιμότητα: <https://www.statista.com/statistics/750328/worldwide-smartwatch-market-share-by-platform/>
- [23] Huawei Watch 2 Sport Specifications - https://www.gsmarena.com/huawei_watch_2-8585.php
- [24] “Java” – Διαθεσιμότητα: <https://el.wikipedia.org/wiki/Java>
- [25] “XML” – Διαθεσιμότητα: <https://el.wikipedia.org/wiki/XML>
- [26] Fournier-Viger, P., Lin, C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., Lam, H. T. (2016). [The SPMF Open-Source Data Mining Library Version 2](#). Proc. 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016) Part III, Springer LNCS 9853, pp. 36-40.
- [27] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [28] opencsv – Διαθεσιμότητα: <http://opencsv.sourceforge.net/>
- [29] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Bill Chiu - A Symbolic Representation of Time Series, with Implications for Streaming Algorithms - University of California – Riverside, Computer Science & Engineering Department, Riverside, CA 92521, USA
- [30] Piecewise Aggregate Approximation of time series – Διαθεσιμότητα: https://jmotif.github.io/sax-vsm_site/morea/algorithm/PAA.html

[31] Symbolic Aggregate approXimation – Διαθεσιμότητα:

https://jmotif.github.io/sax-vsm_site/morea/algorithm/SAX.html

ΠΑΡΑΡΤΗΜΑ Α – Οι Διακριτές Χειρονομίες

Οι 5 διακριτές χειρονομίες που είναι σε θέση να αναγνωρίζει το σύστημα είναι αυτές που φαίνονται στις παρακάτω εικόνες:

hh



hu



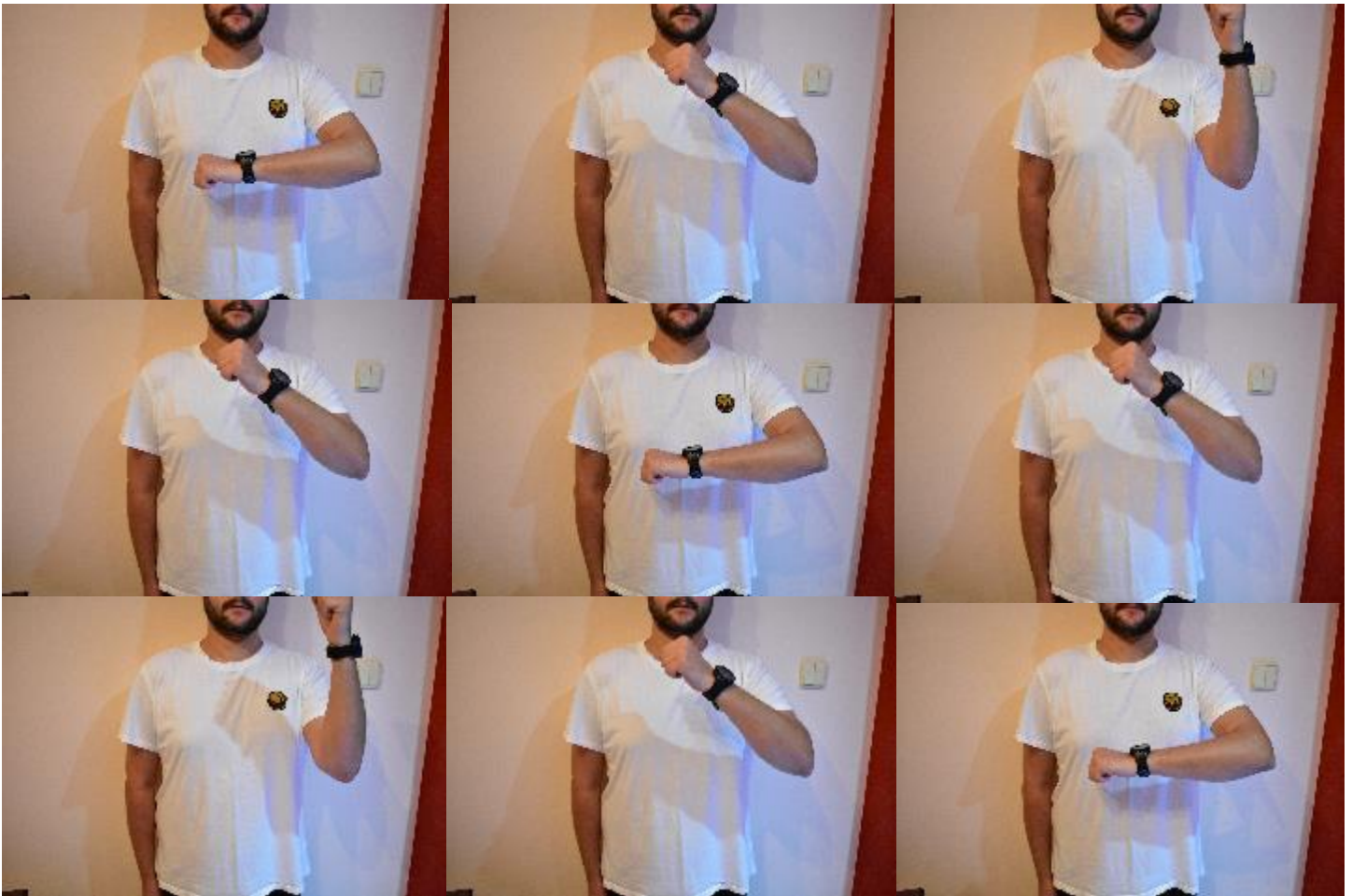
hud



hh2

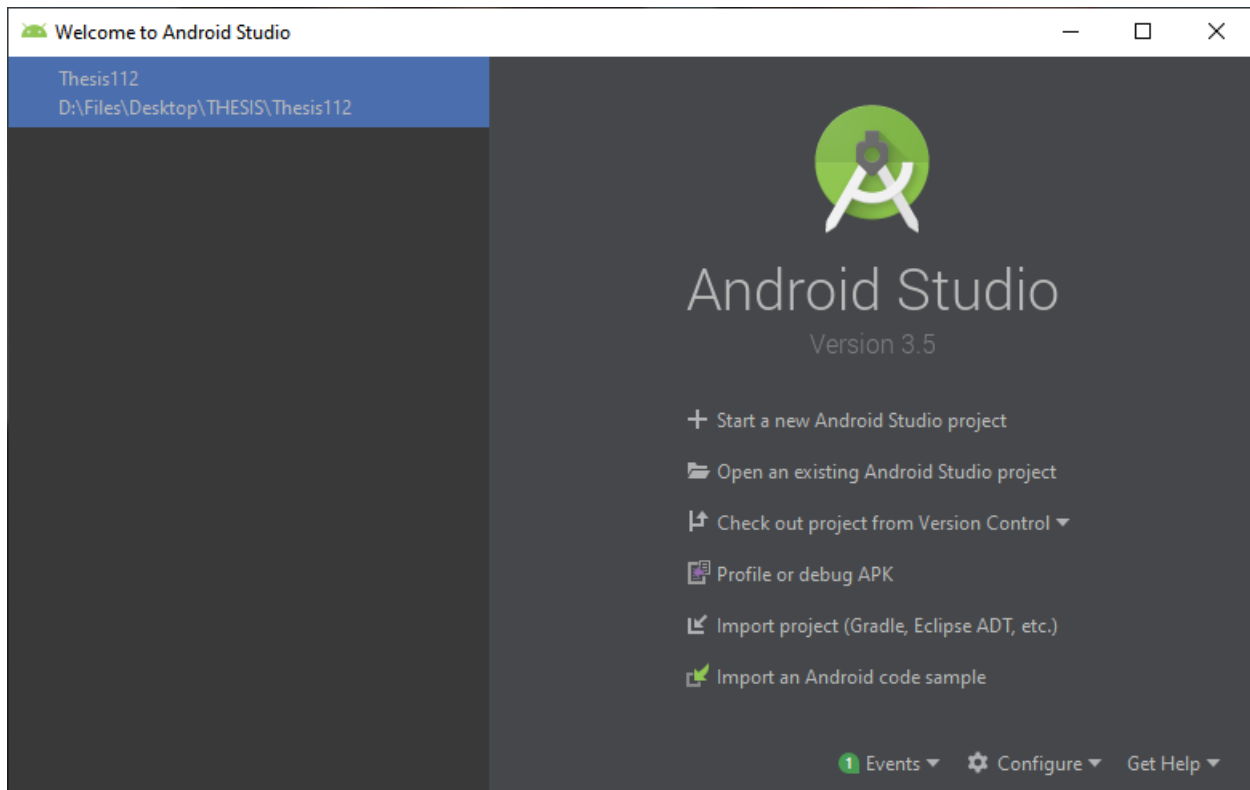


hu2



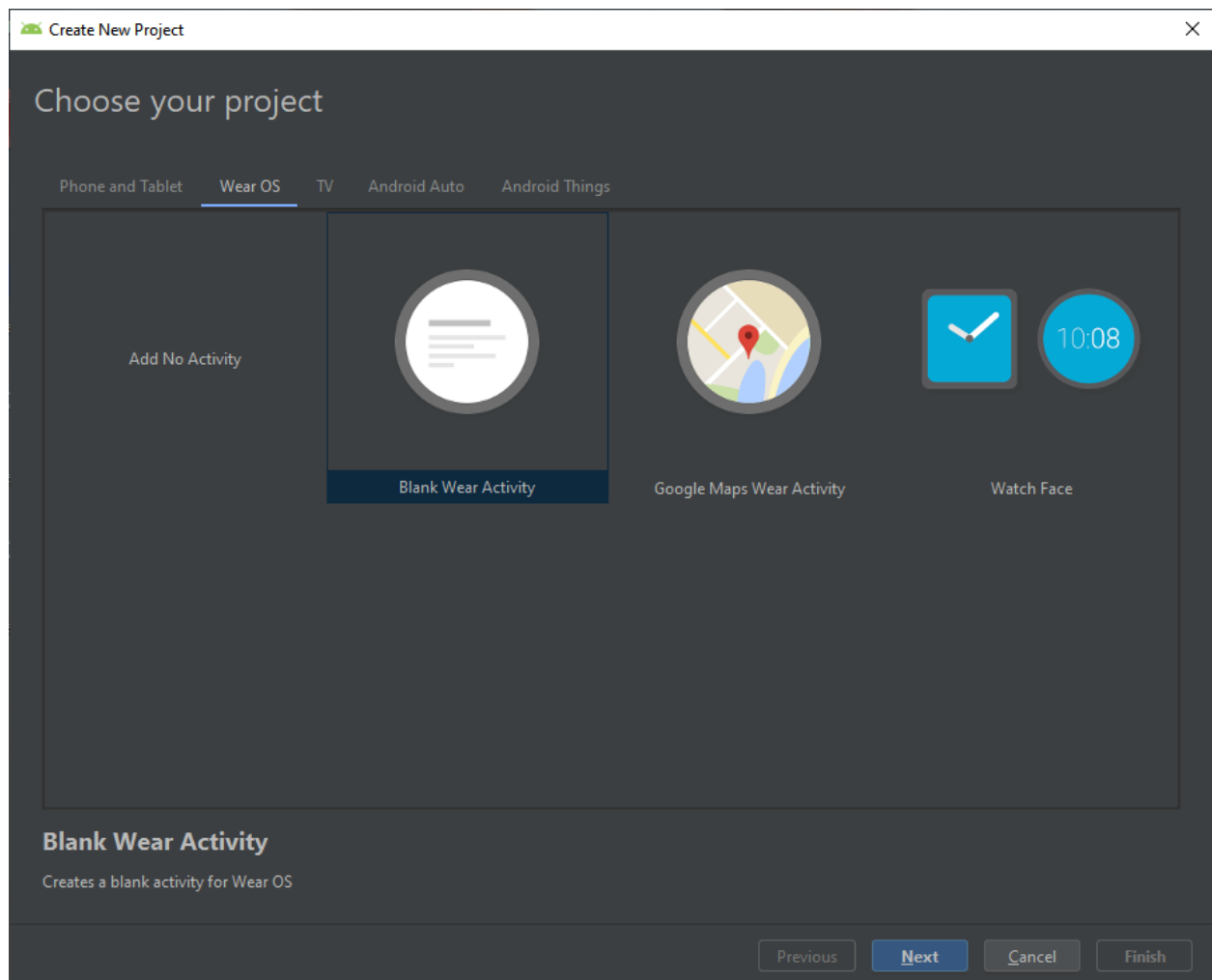
ΠΑΡΑΡΤΗΜΑ Β – Οδηγίες για τη Δημιουργία Project με Wear και Mobile Module στο Android Studio

Για τη δημιουργία ενός Project στο Android Studio που θα περιέχει το mobile και το wear module για την ανάπτυξη κώδικα στο έξυπνο κινητό και στο έξυπνο ρολόι αντίστοιχα ακολουθείται η παρακάτω διαδικασία.

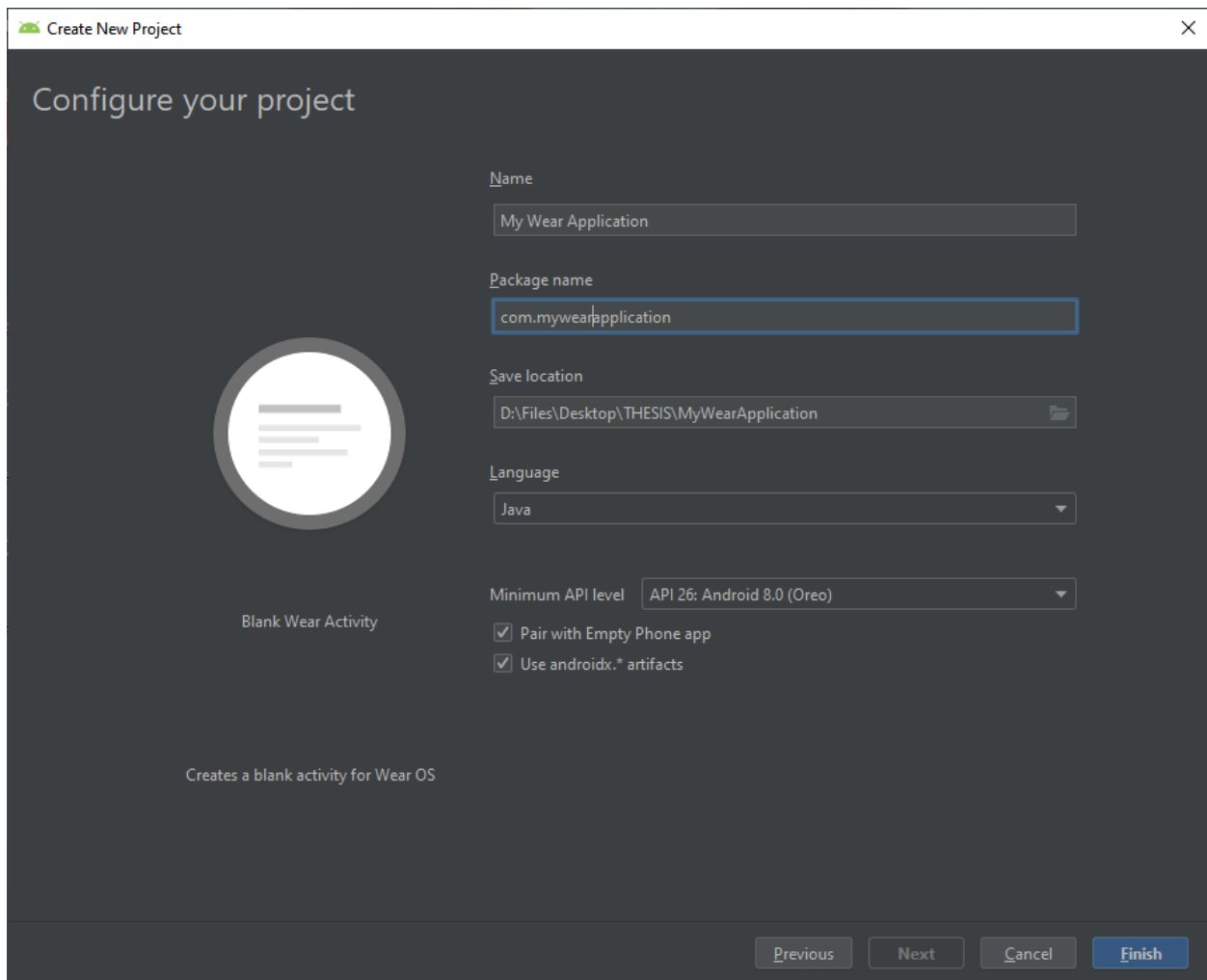


Όταν ανοίξει το παράθυρο του Android Studio στην εκκίνηση επιλέγουμε το εικονίδιο με το σταυρό που γράφει **“Start a new Android Studio Project”**

Στη συνέχεια στο παράθυρο που εμφανίζεται επιλέγεται η δεύτερη καρτέλα που γράφει **“Wear OS”** και η επιλογή **“Blank Wear Activity”** και μετά **“Next”**



Στο επόμενο παράθυρο που εμφανίζεται συμπληρώνεται το όνομα της εφαρμογής στο πεδίο **“Name”**, το όνομα του πακέτου στο πεδίο **“Package name”** και η τοποθεσία για αποθήκευση του project στον υπολογιστή στο πεδίο **“Save Location”**. Στη συνέχεια επιλέγεται η γλώσσα προγραμματισμού **“Java”** από το πεδίο **“Language”** και το **“minimum API Level”** αντιστοιχεί στο API Level που υποστηρίζει η φορητή υπολογιστικής ικανότητας. Τέλος τα επόμενα δύο πεδία πρέπει να επιλεγούν για τη δημιουργία του Mobile Module παράλληλα με τη δημιουργία του Wear στο project και να πατηθεί το κουμπί **“Finish”**.



Create New Project

Configure your project

Name
My Wear Application

Package name
com.mywearapplication

Save location
D:\Files\Desktop\THESIS\MyWearApplication

Language
Java

Minimum API level
API 26: Android 8.0 (Oreo)

☒ Pair with Empty Phone app
☒ Use androidx.* artifacts

Blank Wear Activity

Creates a blank activity for Wear OS

Previous Next Cancel Finish

Το επόμενο παράθυρο που θα εμφανιστεί είναι ο Editor του Android Studio και μετά από λίγη αναμονή μέχρι να γίνει το αρχικό build του project το πρόγραμμα είναι έτοιμο για χρήση.

