**Πανεπιστήμιο Αθηνών**

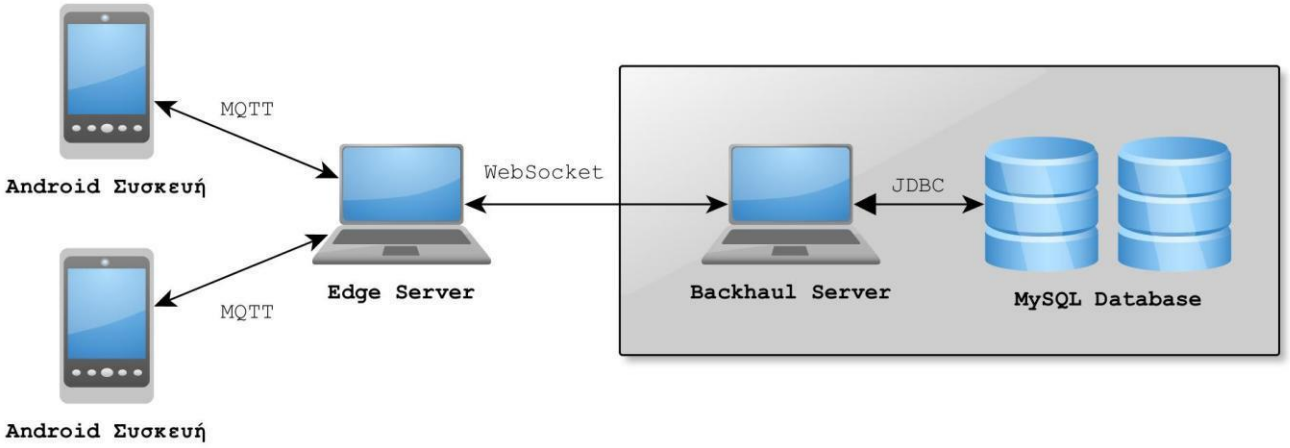
**Τμήμα Πληροφορικής και Τηλεπικοινωνιών**

**Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες**

Χειμερινό εξάμηνο
2018 – 2019

Σκοπός αυτής της εργασίας είναι η ανάπτυξη ενός συστήματος αποφυγής σύγκρουσης μέσα στο χώρο.

Στο σχήμα που ακολουθεί γίνεται μία γραφική απεικόνιση της συνολικής αρχιτεκτονικής που πρέπει να αναπτυχθεί στα πλαίσια της εργασίας. Επιπρόσθετα, γίνεται αναλυτική εξήγηση των δομικών στοιχείων που απαρτίζουν την αρχιτεκτονική.




- **<u>Android Συσκευή</u>**
  Πρόκειται για την κινητή συσκευή, η οποία είναι υπεύθυνη για την συγκέντρωση μετρήσεων από τους εγκατεστημένους αισθητήρες της (όπως **accelerometer**), τιμές οι οποίες θα αποστέλλονται στον **Edge Server** με τη χρήση του **MQTT** πρωτοκόλλου. Επιπρόσθετα, οι κινητές συσκευές θα αποστέλλουν στον **Edge Server** το στίγμα **GPS** τους για την εύρεση της θέσης τους μέσα στον χώρο. Τέλος, οι κινητές συσκευές θα αποστέλλουν σε πραγματικό χρόνο τα **EEG** δεδομένα που έχουν συγκεντρωθεί για τις ανάγκες της εργασίας. Συνεχίζοντας, η επικοινωνία των **Android Συσκευών** με το **Edge Server** είναι **αμφίδρομη** καθώς θα λαμβάνουν ειδοποιήσεις σε τακτά χρονικά διαστήματα σχετικά με την **κατάσταση** τους. Οι εντολές κατάστασης των τερματικών περιγράφονται αναλυτικά παρακάτω.
- **<u>Edge Server</u>**
  Ο **Edge Server** είναι υπεύθυνος για τη συλλογή των δεδομένων που αποστέλλονται από τα τερματικά καθώς και για την αποστολή των εντολών κατάστασης προς αυτά. Ένας ακόμα ρόλος του **Edge Server** είναι το **classification** των **EEG** δεδομένων που

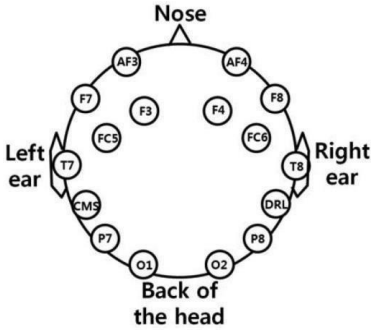
προέρχονται από τις κινητές συσκευές με σκοπό την εύρεση εντολών κατάστασης. Η κατάσταση η οποία εξάχθηκε από το **classification** πέρα από την αποστολή της στις κινητές συσκευές, αποστέλλεται και στο **Backhaul Server** η περιγραφή του οποίου δίνεται στην επόμενη γραμμή.

- **Backhaul Server**
  Ο **Backhaul Server** αποτελεί ακόμα ένα δομικό στοιχείο της εργασίας καθώς είναι υπεύθυνος για το **training** του μοντέλου που θα χρησιμοποιηθεί από τον **Edge Server** για το **classification**. Όπως γίνεται εύκολα κατανοητό μετά το **training** του μοντέλου, αυτό θα πρέπει να αποσταλεί στον **Edge Server** με τρόπο ο οποίος θα αναλυθεί στις παρακάτω σελίδες. Τέλος, ο **Backhaul Server** είναι υπεύθυνος για τη λήψη των εντολών κατάστασης των τερματικών από τον **Edge Server** (**όπως αναφέρθηκε και στην περιγραφή του Edge Server**) και για την αποθήκευσή τους σε μία **MySQL** βάση δεδομένων με σκοπό τη διατήρηση ιστορικού.
- **MySQL Database**
  Το τελευταίο δομικό στοιχείο της εργασίας είναι η **MySQL** βάση δεδομένων, η οποία θα διατηρεί το ιστορικό κατάστασης των τερματικών συσκευών. Οι καταστάσεις αυτές αποστέλλονται όπως έχει ήδη αναφερθεί από το **Backhaul Server**.

Όλα τα **EEG** δεδομένα που παρέχονται από τη φορητή συσκευή ηλεκτροεγκεφαλογραφήματος **EMOTIV Epoc+** σας παρέχονται υπό την μορφή **CSV** αρχείων. Οι τιμές των **CSV** αρχείων ακολουθούν την παρακάτω δομή:
Τα πρώτα **δεκατέσσερα** στοιχεία αντιπροσωπεύουν τις τιμές των **14** αισθητήρων (**channels**) και τα οποία είναι:

**AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4**

Η γραφική απεικόνιση της θέσης του κάθε αισθητήρα είναι αυτή που παρουσιάζεται στην παρακάτω Εικόνα.



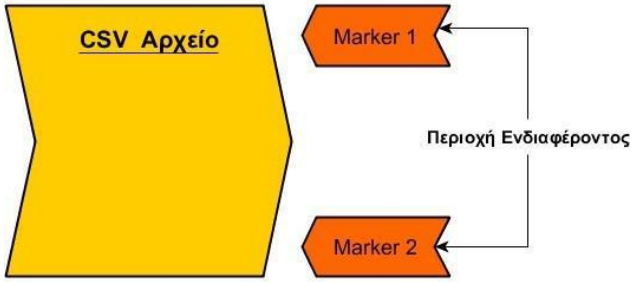

Τα επόμενα **δεκατέσσερα** στοιχεία αντιπροσωπεύουν την ποιότητα σύνδεσης των **14** αισθητήρων με το ανθρώπινο κεφάλι κατά τη διάρκεια συλλογής δεδομένων και τα οποία είναι:

**CQ_AF3, CQ_F7, CQ_F3, CQ_FC5, CQ_T7, CQ_P7, CQ_O1, CQ_O2, CQ_P8, CQ_T8, CQ_FC6, CQ_F4, CQ_F8, CQ_AF4**

Οι δυνατές τιμές της ποιότητας σύνδεσης είναι **0, 1, 2, 3, 4** με το **0** να συμβολίζει την αδυναμία επαφής και το **4** τη βέλτιστη ποιότητα σύνδεσης.

Όσον αφορά τα τελευταία δύο στοιχεία ισχύουν τα εξής:

Ο **Marker** είναι η μεταβλητή που δηλώνει την έναρξη ή τη λήξη κάποιου **event / περιοχής ενδιαφέροντος** (π.χ σε αυτό το σημείο ξεκίνησε ο χρήστης να κλείνει τα μάτια του), ορίζεται μέσω συντόμευσης κατά τη φάση της δειγματοληψίας. Ο **MarkerH** είναι μεταβλητή με τιμή που καθορίζει με ποιο **Hardware Input** (π.χ πληκτρολόγιο, ποντίκι) ορίστηκε ο **Marker**. Στα **CSV** αρχεία που περιέχουν τις τιμές δειγματοληψίας, μεταξύ δύο μη μηδενικών **Marker** λαμβάνει χώρα κάποιο **event**, όπως κλειστά ή ανοιχτά μάτια.
Μία γραφική απεικόνιση της λειτουργίας των **Marker** φαίνεται στην παρακάτω εικόνα.




## Δεύτερη Φάση της Εργασίας

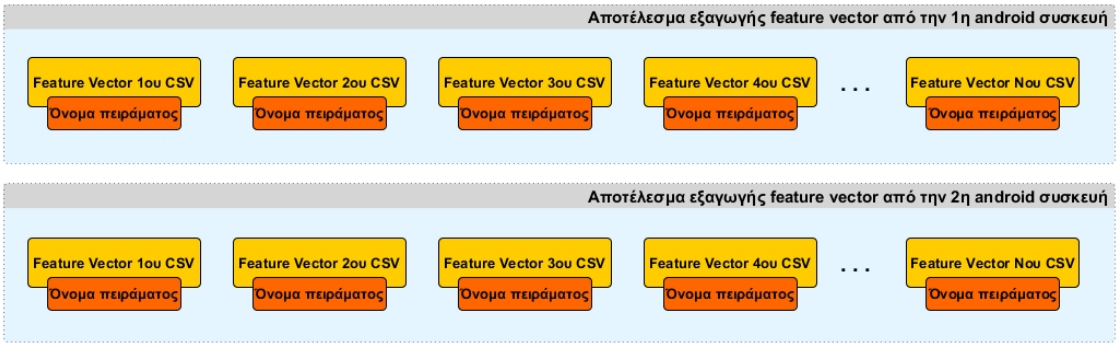
Στη δεύτερη φάση της εργασίας καλείστε να επεκτείνετε και να ολοκληρώσετε την εφαρμογή που υλοποιήθηκε στην πρώτη φάση. Πιο συγκεκριμένα, τα android τερματικά θα **εκτελούν** τις εντολές που **παρουσιάστηκαν** στο πρώτο παραδοτέο, οι οποίες θα έρχονται από τον **Edge Server** σύμφωνα με την κατηγοριοποίηση που θα εξηγηθεί στις επόμενες σελίδες. Επιπρόσθετα, ο **Edge Server** με την σειρά του πέρα από κατηγοριοποίηση, θα κάνει παρουσίαση στατιστικών στοιχείων σχετικά με την επιτυχία του αλγόριθμου κατηγοριοποίησης. Τέλος, θα στέλνει αρχεία **log** πίσω στον Backhaul Server, ο οποίος θα αναλαμβάνει την εναπόθεσή τους στην βάση δεδομένων που έχει παρουσιαστεί και θα εξηγηθεί περεταίρω σε αυτήν την φάση. Παρακάτω, γίνεται αναλυτική παρουσίαση των δομικών στοιχείων που **χρειάζονται επέκταση** σε αυτή τη φάση της εργασίας.

- **Edge Server**
  Σε αυτήν την φάση της εργασίας ο Edge Server θα πρέπει να έχει ήδη όλες τις διασυνδέσεις και τις δομές που περιγράφτηκαν στο πρώτο παραδοτέο. Η επέκταση του θα αφορά την κατηγοριοποίηση των δειγμάτων και την παρουσίαση στατιστικών πληροφοριών σχετικά με την απόδοση του αλγορίθμου κατηγοριοποίησης. Για την υλοποίηση του αλγορίθμου κατηγοριοποίησης ο **Edge Server** θα πρέπει να **εξάγει** το **feature vector** για **κάθε** ένα από τα **CSV** αρχεία που λαμβάνει από τα **τερματικά android**. Ο τρόπος **εξαγωγής** αυτού του **feature vector** είναι **ίδιος** με αυτόν που χρησιμοποίησε ο **Backhaul Server** στην πρώτη φάση της εργασίας. Οπότε και πάλι, μετά

την ολοκλήρωση της εξαγωγής των **feature vector** **κάθε** **CSV** αρχείου θα πρέπει να είναι ένα διάνυσμα της μορφής:


**Προσοχή!**

Κάθε **feature vector** που εξάγεται θα πρέπει να συνοδεύετε και από το **όνομα** του τερματικού android από το οποίο στάλθηκε το αρχικό **CSV**. Οπότε σε τελική ανάλυση θα θέλαμε μία δομή που ακολουθεί σε γενικές γραμμές την παρακάτω μορφή:




Συνεχίζοντας, κάθε φορά που υπολογίζεται ένα καινούργιο **feature vector**, αυτό θα πρέπει να κατηγοριοποιηθεί σύμφωνα με μία παραλλαγή του **kNN** αλγορίθμου η οποία παρουσιάζεται υπό την μορφή ψευδο-κώδικα παρακάτω.

## Ψευδο-κώδικας παραλλαγής του kNN Αλγορίθμου

Απώτερος σκοπός του αλγορίθμου είναι δοθέντος ενός διανύσματος **x** να βρεθούν οι **k** κοντινότεροι γείτονές τους. Το διάνυσμα **x** θα κατηγοριοποιηθεί στην κλάση στην οποία ανήκει η πλειοψηφία των **k** γειτόνων του.

Έστω **X** τα ήδη κατηγοριοποιημένα δεδομένα(**training set από 1η φάση**), **Y** τα **Labels** για τις κλάσεις (στην συγκεκριμένη εφαρμογή υπάρχουν μόνο 2 κλάσεις με ονόματα **EyesClosed** και **EyesOpened**) και **x** το διάνυσμα προς κατηγοριοποίηση και m το μέγεθος των κατηγοριοποιημένων δεδομένων.

**For** $\mathrm{i} = 1$ to **m** do
Υπολογισμός *EuclideanDistance* $\mathbf{d}(\mathbf{X_i}, \mathbf{x})$
**End** for
**Υπολογισμός** του συνόλου **l** μέσα στο οποίο θα βρίσκονται οι **k** μικρότερες αποστάσεις $\mathbf{d}(\mathbf{X_i}, \mathbf{x})$
**Υπολογισμός counter** που δηλώνει τον αριθμό των $\mathbf{Y_i}$ μέσα στο **l**
**Υπολογισμός** της μετρικής βάρους $\mathbf{w_{y_i}} = \mathbf{w_{y_i}} + \mathbf{1/d}(\mathbf{X_i}, \mathbf{x})$ για κάθε $\mathbf{X_i} \in$ στο σύνολο των **k** γειτόνων με **Label** $\mathbf{Y_i}$
Επιστροφή του $\mathbf{Y_i}$ για το οποίο το $\mathbf{w_{y_i}} * \mathbf{counter}$ παίρνει την μεγαλύτερη τιμή.

**Σημείωση!**

Η επιλογή του **k** είναι ανοιχτή. Συνήθως οι τιμές του είναι περιττοί αριθμοί για να αποφεύγονται οι ισοψηφίες. Συνήθεις τιμές είναι 3,5,7,11.

Στην συνέχεια, θα γίνει αναλυτική εξήγηση σχετικά με την αποστολή εντολών πίσω στα **android** τερματικά. Όπως αναφέρθηκε και στην πρώτη φάση της εργασίας τα είδη των εντολών είναι **2**. Για υπενθύμιση παραθέτονται παρακάτω:

- Το **1ο είδος εντολής** θα αφορά κίνδυνο που σχετίζεται αποκλειστικά και μόνο με το συγκεκριμένο **android** τερματικό. Σε αυτήν την περίπτωση το **android** τερματικό θα πρέπει να εκτελεί μία ηχητική λειτουργία της επιλογής σας.
- Το **2ο είδος εντολής** αφορά τον πιθανό κίνδυνο του συγκεκριμένου **android** τερματικού από κάποιο άλλο **android** τερματικό το οποίο βρίσκεται σε κίνδυνο. Σε αυτήν την περίπτωση πέρα από ηχητικό μήνυμα θα πρέπει να γίνεται και ενεργοποίηση του **flash** της κάμερας του κινητού.

Τα σενάρια των διαφορετικών εντολών προς τα **android** τερματικά παρουσιάζονται παρακάτω: (**Προσοχή!!** Μας ενδιαφέρει να εξετάσουμε την **κατηγοριοποίηση** των δειγμάτων και **όχι** την πραγματική κλάση τους. Για παράδειγμα εάν σταλεί ένα **eyes closed CSV** από το τερματικό **android** αλλά η κατηγοριοποίηση **λανθασμένα** το κατατάξει στην κλάση **eyes opened** τότε εμείς θα το χειριστούμε ως **eyes opened**.)

1. Εάν γίνει κατηγοριοποίηση **3ων** συνεχόμενων **Eyes Closed** από το ίδιο **android** τερματικό τότε έχουμε ενεργοποίηση του **1ου** είδους εντολής. Την ηχητική ειδοποίηση θα πρέπει να συνοδεύει προειδοποιητικό μήνυμα για τον οδηγό του οχήματος.
   Για κάθε **επόμενο Eyes Closed** από το **ίδιο android** τερματικό και **μέχρις ότου** γίνει κατηγοριοποίηση **Eyes Opened** η ηχητική ειδοποίηση και το προειδοποιητικό μήνυμα του 1ου είδους εντολής παραμένουν ενεργοποιημένα.
2. Εάν γίνει κατηγοριοποίηση **3ων** ή περισσοτέρων συνεχόμενων **Eyes Closed** από ένα **android** τερματικό και ο **Edge Server** διαπιστώσει ότι το συγκεκριμένο **android** τερματικό βρίσκεται σε κοντινή απόσταση από ένα **2ο android** τερματικό **τότε** έχουμε ενεργοποίηση του **2ου** είδους εντολής και για τα **2** τερματικά μαζί με προειδοποιητικό μήνυμα. Το μέγιστο όριο απόστασης δύο τερματικών βρίσκεται στην ελευθερία σας. Ο υπολογισμός απόστασης με χρήση **GPS** στίγματος μπορεί να βρεθεί **online**.

Σε περίπτωση που ισχύουν και τα δύο παραπάνω σενάρια η ηχητική ειδοποίηση και το φλάς παραμένουν ενεργά και το προειδοποιητικό μήνυμα τροποποιείται καταλλήλως ώστε να καλύπτει και τα δύο σενάρια.

Κάθε ένα από τα παραπάνω σενάρια μόλις εντοπίζεται θα πρέπει να μπαίνει υπό την μορφή της συγκεκριμένης εντολής μέσα σε ένα **ενταμιευτή-buffer** ώστε να σταλθεί στο/στα τερματικό/α **android**. Παράδειγμα μετάφρασης των σεναρίων σε εντολές είναι:

1. "Execute Eyes Closed Single Danger Level 1"
2. "Execute Eyes Closed Double Danger Level 2"

Σχετικά με την αποστολή των εντολών θα πρέπει να **δημιουργηθεί ένα thread** το οποίο θα είναι αποκλειστικά υπεύθυνο μόνο για αυτή την αποστολή τους στα **android** τερματικά. Ακόμα, θα **πρέπει να υπάρχει ακόμα ένα thread** το οποίο θα πρέπει να στείλει τις εντολές **πίσω στον backhaul server** ώστε να εναποθετηθούν στην βάση δεδομένων ως **log** αρχεία.

Εδώ έρχεται το **criticality level** που είχε παρουσιαστεί στο **τέλος της 1ης φάσης** της εργασίας. Όπως είδαμε παραπάνω υπάρχουν 2 διαφορετικά σενάρια οπότε θα έχουμε 2 διαφορετικά **criticality levels**. Αυτά θα είναι:

1. "Execute Eyes Closed Single Danger Level 1" - **Criticality Level 1**
2. "Execute Eyes Closed Double Danger Level 2" - **Criticality Level 2**

Στην συνέχεια για λόγους σαφήνειας ακολουθούν 2 **παραδείγματα** αποθήκευσης στην βάση για κάθε ένα από τα πιθανά σενάρια.

| Αναγνωριστικό Android Συσκευής | Timestamp (Date&Time) | GPS Signal | Criticality Level |
|---|---|---|---|
| Android 1 ID | 01/01/2019/15:30 | 37.9684328,23.7647199 | 1 |
| Android 1 ID__ Android 2 ID | 01/01/2019/17:32 | 37.9684328,23.7647199__37.9682908,23.7666628 | 2 |

**Προσοχή!!**
Κάθε εντολή αφού αποσταλεί στα **android** τερματικά καθώς και στον **backhaul server** θα πρέπει να αφαιρείται από τον **ενταμιευτή-buffer**. Εδώ πέρα σημαντικό είναι να τονιστεί ότι θα πρέπει να υπάρχει συγχρονισμός για την πρόσβαση στον **buffer**.

Τέλος, θα πρέπει να γίνουν μερικές ακόμα επεκτάσεις της **java εφαρμογής** στον **Edge Server** που έχουν ως σκοπό την παρακολούθηση την επιτυχίας του αλγορίθμου κατηγοριοποίησης. Πιο συγκεκριμένα:

- Θα πρέπει να γίνεται διασταύρωση μεταξύ της κλάσης στην οποία τελικά κατηγοριοποιήθηκε το δείγμα και της πραγματικής κλάσης του δείγματος (η οποία όπως έχει αναφερθεί μπορεί να παρθεί από το όνομα του αρχείου). Στην **κονσόλα** της **java** εφαρμογής θα πρέπει **κάθε** φορά που γίνεται **κατηγοριοποίηση** ενός **feature vector** να τυπώνεται η κλάση στην οποία τελικά κατηγοριοποιήθηκε το δείγμα καθώς

και η πραγματική κλάση του δείγματος μαζί με ένα μήνυμα επιτυχίας ή αποτυχίας. Η επιτυχία θα είναι εάν η πραγματική κλάση είναι ίδια με την κλάση στην οποία κατηγοριοποιήθηκε το δείγμα και η αποτυχία θα προκύπτει από το αν τελικά οι 2 κλάσεις ήταν διαφορετικές.

- ❖ Με την ολοκλήρωση ή τον τερματισμό του προγράμματος θα πρέπει να εμφανίζεται ένα συνολικό ποσοστό επιτυχίας του αλγορίθμου κατηγοριοποίησης. Το ποσοστό επιτυχίας ορίζεται από το πόσα δείγματα τελικά κατηγοριοποιήθηκαν στις σωστές κλάσεις.

- **Backhaul Server**
  Ο σκοπός του **Backhaul S**erver σε αυτήν την φάση της εργασίας είναι απλός. Θα πρέπει να παραλαμβάνει τα σενάρια σχετικά με το **criticality level** που εξηγήθηκε αναλυτικά παραπάνω και να τα αποθηκεύει στην βάση δεδομένων ως ιστορικό.

Επιπρόσθετα σε αυτό το παραδοτέο θα πρέπει να παραδοθεί και ένα αρχείο **README** το οποίο θα περιέχει αναλυτικά τον τρόπο λειτουργίας του προγράμματος.

**Τεχνολογίες συστήματος ανάπτυξης:**

1. Java Oracle SE 8
2. Android API 4.1 ή νεότερο
3. Android Studio (Εργαλείο ανάπτυξης της εφαρμογής Android)
4. IntelliJ IDEA (Εργαλείο ανάπτυξης της Java εφαρμογής)
5. MQTT mosquitto broker
6. Eclipse Paho (Java & Android MQTT client)

Κατά τη διαδικασία της ανάπτυξης της εργασίας είναι η υποχρεωτική η χρήση του εργαλείου Git (Version Control) και της πλατφόρμας gitlab (anapgit.scanlab.gr) που παραχωρείται για τις ανάγκες του μαθήματος.

Η υλοποίηση της εφαρμογής θα πρέπει:

1. Να υπακούει στις αρχές του αντικειμενοστραφούς προγραμματισμού
2. Να γίνεται σωστή και αποδοτική οργάνωση του κώδικα σε κλάσεις και πακέτα.
3. Να είναι όσο το δυνατό παραμετροποιήσιμη και δυναμική γίνεται.
4. Να γίνεται σωστή και αποδοτική διαχείριση της μνήμη.