



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Χαροκόπειο Πανεπιστήμιο
Τμήμα Πληροφορικής και Τηλεματικής

Αλγόριθμοι μέτρησης φυσικής δραστηριότητας από αισθητήρες κινητού τηλεφώνου

Πτυχιακή εργασία
ΚΙΤΣΑΚΗΣ ΓΕΩΡΓΙΟΣ
Α.Μ. 219127

Αθήνα, 2023



HAROKOPIO UNIVERSITY

Harokopio University
Department Informatics and Telematics

Algorithms for physical activity measurement from smartphone sensors

KITSAKIS GEORGIOS

I.T. 219127

Athens, 2023

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Επιβλέπων

ΔΙΟΥ ΧΡΗΣΤΟΣ

Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Μέλη

ΜΠΑΡΔΑΚΗ ΚΛΕΟΠΑΤΡΑ

Επίκουρη Καθηγήτρια, Τμήμα Πληροφορικής και τηλεματικής, Χαροκόπειο Πανεπιστήμιο

ΔΗΜΗΤΡΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ

Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Ο Κιτσάκης Γεώργιος δηλώνω υπεύθυνα ότι:

1. Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.
2. Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.
3. Όπου υφίστανται δικαιώματα άλλων δημιουργών έχουν διασφαλιστεί όλες οι αναγκαίες άδειες χρήσης ενώ το αντίστοιχο υλικό είναι ευδιάκριτο στην υποβληθείσα εργασία.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πολύ τον κύριο Διου για την εμπιστοσύνη που μου έδειξε αλλά και την πολύτιμη καθοδήγηση του για την επίτευξη της πτυχιακής μου εργασίας. Ευχαριστώ επίσης την οικογένειά μου και τους φίλους μου για την συμπαράσταση και την υπομονή τους καθ' όλη τη διάρκεια φοίτησής μου στο Χαροκόπειο Πανεπιστήμιο.

Περιεχόμενα

1	Εισαγωγή	16
1.1	Περιγραφή προβλήματος	16
1.2	Στόχος της πτυχιακής εργασίας	16
2	Θεωρητικό υπόβαθρο και βιβλιογραφική επισκόπηση	17
2.1	Μηχανική μάθηση και νευρωνικά δίκτυα	17
2.2	Συνελκτικά νευρωνικά δίκτυα	17
2.3	Αισθητήρες	18
2.3.1	Επιταχυνσόμετρο	19
2.3.2	Μαγνητόμετρο	19
2.3.3	Γυροσκόπιο	20
2.4	Βιβλιογραφική επισκόπηση	21
3	Αλγόριθμοι μέτρησης της κίνησης	22
3.1	Περιγραφή του Συνόλου Δεδομένων και της Προεπεξεργασίας δεδομένων	22
3.1.1	Σύνολο Δεδομένων	22
3.1.2	Προεπεξεργασία δεδομένων	23
3.2	Περιγραφή του σετ εκπαίδευσης, δοκιμής και επικύρωσης	24
3.3	Συναρτήσεις Ενεργοποίησης	25
3.3.1	Ανορθωμένη Γραμμική Μονάδα – ReLU	25
3.3.2	Συνάρτηση SoftMax	26
3.4	Τύποι υπερ-παραμέτρων	27
3.5	Adam Optimizer	28
3.6	Μετρικές	28
3.6.1	Ευστοχία (Accuracy)	28
3.6.2	Ακρίβεια (Precision)	28

3.6.3	Ανάκληση (Recall)	29
3.6.4	F1 score	29
4	Documentation του Κώδικα	30
4.1	Preprocessing	30
4.2	Data Handler	31
4.3	CNN – CNN + LSTM - CNN + GRU	31
4.4	Load Model	34
5	Αισθητήρας στη θέση του χεριού – Πείραμα 1	37
5.1	Αρχιτεκτονική των μοντέλων	37
5.2	Μοντέλα με έναν αισθητήρα	37
5.2.1	Μοντέλο CNN	37
5.2.2	Μοντέλο CNN-LSTM	39
5.2.3	Μοντέλο CNN-GRU	40
5.3	Μοντέλα με τρεις αισθητήρες	41
5.3.1	Μοντέλο CNN	41
5.3.2	Μοντέλο CNN-LSTM	42
5.3.3	Μοντέλο CNN-GRU	43
5.4	Αποτελέσματα	43
5.4.1	Αποτελέσματα από έναν αισθητήρα	44
5.4.2	Αποτελέσματα από τρεις αισθητήρες	44
5.4.3	Επιπλέον μετρικές	45
5.4.4	Σύγκριση αποτελεσμάτων	45
6	Αισθητήρες στα υπόλοιπα σημεία – Πείραμα 2	46
6.1	Πρώτος χρήστης	46
6.1.1	Αισθητήρες στο χέρι	46
6.1.2	Αισθητήρες στην τσάντα	47

6.1.3	Αισθητήρες στο στήθος	48
6.1.4	Αισθητήρες στην πρόσθια τσέπη	48
6.2	Δεύτερος χρήστης	49
6.2.1	Αισθητήρες στο χέρι	49
6.2.2	Αισθητήρες στην τσάντα	49
6.2.3	Αισθητήρες στο στήθος	50
6.2.4	Αισθητήρες στην πρόσθια τσέπη	50
6.3	Τρίτος χρήστης	51
6.3.1	Αισθητήρες στο χέρι	51
6.3.2	Αισθητήρες στην τσάντα	51
6.3.3	Αισθητήρες στο στήθος	52
6.3.4	Αισθητήρες στην πρόσθια τσέπη	52
6.4	Σύγκριση αποτελεσμάτων	53
6.4.1	Πρώτος χρήστης	53
6.4.2	Δεύτερος χρήστης	53
6.4.3	Τρίτος χρήστης	53
7	Leave-one-subject-out	54
7.1	Έννοια	54
7.2	Παρόν πείραμα	54
7.3	Αποτελέσματα χρηστών	61
7.3.1	Χρήστης 1	61
7.3.2	Χρήστης 2	62
7.3.3	Χρήστης 3	63
7.4	Συμπέρασμα	63
8	Η χρήση του ChatGPT	64
9	Συμπεράσματα και μελλοντικές προεκτάσεις	65

Περίληψη

Η αναγνώριση της δραστηριότητας του ανθρώπου είναι ένα σημαντικό πρόβλημα στην πεδιάδα της αναγνώρισης προτύπων. Η επιτυχής επίλυση αυτού του προβλήματος μπορεί να βοηθήσει στη βελτίωση της ποιότητας ζωής των ανθρώπων, όπως στη βελτίωση της φροντίδας της υγείας και της ανεξαρτησίας των ηλικιωμένων.

Το Sussex Locomotion Dataset είναι ένα σύνολο δεδομένων που περιλαμβάνει καταγραφές κινήσεων από αισθητήρες επάνω σε φορητές συσκευές. Τα δεδομένα που παρέχονται περιλαμβάνουν καταγραφές κινήσεων. Η πτυχιακή εργασία αυτή αποσκοπεί στην ανάπτυξη μιας μεθόδου που θα επιτρέπει την αναγνώριση δραστηριοτήτων από το Sussex Locomotion Dataset. Συγκεκριμένα, ο στόχος είναι να αναπτυχθεί ένα μοντέλο μηχανικής μάθησης που θα μπορεί να προβλέπει τη δραστηριότητα του χρήστη κάθε 30 δευτερολέπτων.

Θα βασίζεται σε αλγόριθμους μηχανικής μάθησης και θα αξιοποιεί τα δεδομένα που παρέχει το Sussex Locomotion Dataset. Η καταγραφή της έρευνας μπορεί να βοηθήσει σε πολλούς τομείς, όπως στην υγεία, την ασφάλεια και την εκπαίδευση.

Abstract

The recognition of human activity is an important problem in the field of pattern recognition. Successfully solving this problem can help improve the quality of people's lives, such as enhancing healthcare and the independence of the elderly.

The Sussex Locomotion Dataset is a collection of motion recordings from sensors on portable devices. The provided data includes motion records. This thesis aims to develop a method that allows the recognition of activities from the Sussex Locomotion Dataset. Specifically, the goal is to develop a machine learning model capable of predicting the user's activity every 30 seconds.

It will be based on machine learning algorithms and will leverage the data provided by the Sussex Locomotion Dataset. The research findings can be helpful in various fields, such as healthcare, security, and education.

Κατάλογος σχημάτων

1	Αρχιτεκτονική δικτύου CNN[15]	17
2	Αισθητήρες σε κινητό[20]	18
3	Πως είναι στους σένσορες τα X,Y,Z[18]	20
4	Label.txt μορφή	23
5	Dataset preview[11][23]	24
6	Dataset plot[11][23]	24
7	ReLU[27]	25
8	Softmax[7]	26
9	Regularization[1]	27
10	Αρχιτεκτονική Adam optimizer[25]	28
11	Ακρίβεια και ανάκληση[21]	29
12	F1 score[14]	29
13	Αρχιτεκτονική και των 3 μοντελών[3]	37
14	plot CNN-VGG	38
15	plot CNN-LSTM	39
16	plot CNN-GRU	40
17	plot CNN-VGG accuracy	41
18	Plot CNN-VGG scores	41
19	plot CNN-LSTM accuracy	42
20	Plot CNN-LSTM scores	42
21	Plot CNN-GRU accuracy	43
22	Plot CNN-GRU scores	43
23	Plot hand accuracy user1	46
24	Plot hand scores user1	46
25	Plot CNN bag accuracy user1	47

26	Plot CNN bag scores user1	47
27	Plot GRU bag accuracy user1	47
28	Plot GRU bag scores user1	47
29	Plot torso accuracy user1	48
30	Plot torso scores user1	48
31	Plot hips accuracy user1	48
32	Plot hips scores user1	48
33	Plot hand accuracy user2	49
34	Plot hand scores user2	49
35	Plot bag accuracy user2	49
36	Plot bag scores user2	49
37	Plot torso accuracy user2	50
38	Plot torso scores user2	50
39	Plot hips accuracy user2	50
40	Plot hips scores user2	50
41	Plot hand accuracy user3	51
42	Plot hand scores user3	51
43	Plot bag accuracy user3	51
44	Plot bag scores user3	51
45	Plot torso accuracy user3	52
46	Plot torso scores user3	52
47	Plot hips accuracy user3	52
48	Plot hips scores user3	52
49	Plot accuracy user1	61
50	Plot scores user1	61
51	Αποτελέσματα με τον χρήστη 1 εκτός	61
52	Plot accuracy user2	62

53	Plot scores user2	62
54	Αποτελέσματα με τον χρήστη 2 εκτός	62
55	Plot accuracy user3	63
56	Plot scores user3	63
57	Αποτελέσματα με τον χρήστη 3 εκτός	63

Code Listings

1	Preprocessing code	30
2	Data Handler code	31
3	CNN	32
4	CNN-LSTM	33
5	CNN-GTU	33
6	Load Model	34
7	Load Model	35
8	Load Model	36
9	Preprocessing code LOOCV	55
10	Data Handler code LOOCV	56
11	CNN code LOOCV	57
12	CNN code LOOCV	58
13	CNN code LOOCV	59
14	CNN code LOOCV	60

Συντομογραφίες

SHL	Sussex Huawei Locomotion
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
NaN	Not a Number
ReLU	Rectified Linear Unit
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
Conv1D	1D Convolution layer
LOOCV	Leave One Out Cross Validation

1 Εισαγωγή

1.1 Περιγραφή προβλήματος

Οι αλγόριθμοι μέτρησης φυσικής δραστηριότητας αποτελούν λογισμικό που μπορεί να εγκατασταθεί σε φορητές συσκευές, όπως κινητά τηλέφωνα, smartwatch και φορητοί αισθητήρες, και η χρήση τους εξυπηρετεί την παρακολούθηση, καταγραφή και ανάλυση της φυσικής δραστηριότητας ενός χρήστη. Με αυτό τον τρόπο επιτυγχάνεται η ενθάρρυνση του χρήστη να υιοθετήσει συνήθειες που θα βελτιώσουν την υγεία και την ποιότητα της ζωής του. Από τα κυριότερα προβλήματα που καλούνται να επιλύσουν οι αλγόριθμοι αυτοί είναι:

- Καταγραφή και ανάλυση των βημάτων: Τα βήματα και η απόσταση που διανύει ο χρήστης κατά τη διάρκεια της ημέρας καταγράφονται μέσω του αλγορίθμου, ώστε να καθοριστεί το επίπεδο της σωματικής δραστηριότητας του ατόμου.
- Παρακολούθηση επιπέδου ενέργειας: Με στόχο την αύξηση της ενεργητικότητας του χρήστη, ο αλγόριθμος αναγνωρίζει την κατάσταση της φυσικής του δραστηριότητας (περπάτημα, τρέξιμο, οδήγηση κτλ.)
- Καρδιακός ρυθμός: Η καταγραφή του καρδιακού ρυθμού κατά τη διάρκεια των καθημερινών δραστηριοτήτων του ατόμου, είναι ιδιαίτερα χρήσιμη για τη διαπίστωση τυχόν προβλημάτων υγείας του.

Για να επιτευχθεί η αναγνώριση και η κατηγοριοποίηση της φυσικής δραστηριότητας του χρήστη χρησιμοποιούνται σύνολα δεδομένων, ένα από τα οποία είναι το Dataset Sussex Huawei Locomotion, το οποίο προσφέρει δεδομένα που περιλαμβάνουν πληροφορίες σχετικά με τις κινήσεις και τη δραστηριότητα του χρήστη, ώστε να δημιουργηθούν οι αλγόριθμοι μέτρησης φυσικής δραστηριότητας.

1.2 Στόχος της πτυχιακής εργασίας

Η ανάπτυξη αλγορίθμων, οι οποίοι έχουν τη δυνατότητα να αξιοποιούν πληροφορίες από ένα σύνολο δεδομένων (dataset), με σκοπό την αναγνώριση της ύπαρξης μετακίνησης, καθώς και τον τρόπο με τον οποίο αυτή επιτυγχάνεται, κατά τη διάρκεια διαφόρων φυσικών δραστηριοτήτων των χρηστών – με τη χρήση αισθητήρων (επιταχυνσόμετρο, γυροσκόπιο, μαγνητόμετρο) κινητού τηλεφώνου, smartwatch – και τη σύγκρισή τους με τα υπάρχοντα δεδομένα. Μέσω αυτής της διαδικασίας, γίνεται μια ανασκόπηση σε αλγόριθμους μηχανικής μάθησης, οι οποίοι είναι ικανοί να επιτύχουν το μέγιστο αποτέλεσμα.

2 Θεωρητικό υπόβαθρο και βιβλιογραφική επισκόπηση

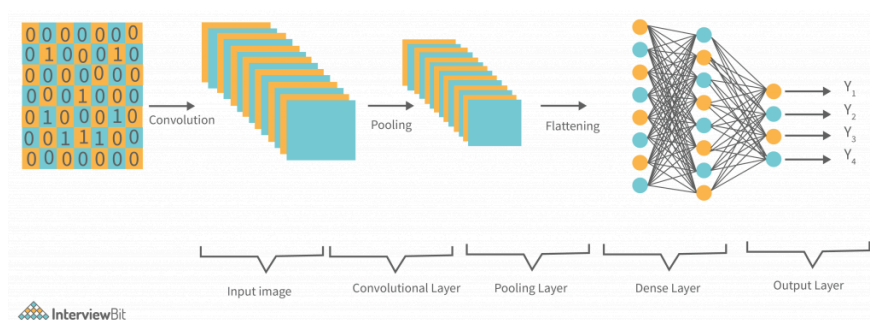
2.1 Μηχανική μάθηση και νευρωνικά δίκτυα

Η μηχανική μάθηση (machine learning)[13] αποτελεί υποκατηγορία της τεχνητής νοημοσύνης (artificial intelligence, AI)[8] και επικεντρώνεται στη δημιουργία αλγορίθμων που έχουν ως στόχο την εκμάθηση νέων δεδομένων από τους υπολογιστές, χωρίς να χρειάζεται ο εκ των προτέρων προγραμματισμός τους. Σκοπός της μηχανικής μάθησης είναι η εξαγωγή πληροφοριών και η πρόβλεψη αποτελεσμάτων για νέα, μη γνωστά δεδομένα. Αυτό έχει ως αποτέλεσμα την αυτοματοποίηση και την ανάπτυξη ευφύων συστημάτων.

Τα νευρωνικά δίκτυα – ένας από τους κυριότερους τύπους μηχανικής μάθησης – συσχετίζονται με τη λειτουργία του ανθρώπινου εγκεφάλου και αντιστοιχούν σε ενεργοποιητές νευρώνων που υπάρχουν σε αυτόν. Η χρήση αλγορίθμων επιτρέπει την εκπαίδευση των νευρωνικών δικτύων, κατά την οποία ελαχιστοποιείται το σφάλμα των προβλέψεών τους συγκριτικά με τις πραγματικές τιμές, προσαρμόζοντας τα βάρη των συνδέσεών τους. Αυτή η διαδικασία είναι επαναλαμβανόμενη έως ότου επιτευχθεί η μέγιστη απόδοση.

2.2 Συνελικτικά νευρωνικά δίκτυα

Παράδειγμα νευρωνικών δικτύων αποτελούν τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks, CNNs)[28], τα οποία μαζί με τα Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks, RNNs) ανήκουν στα βαθιά νευρωνικά δίκτυα. Τα συνελικτικά νευρωνικά δίκτυα προορίζονται κυρίως για την επεξεργασία εικόνων και βίντεο. Βασίζονται στη λειτουργία του οπτικού φλοιού του ανθρώπινου εγκεφάλου και είναι ιδιαίτερα αποδοτικά στο χειρισμό των δομών των εικόνων. Το βασικό χαρακτηριστικό των CNN είναι οι συνελίξεις, οι οποίες εφαρμόζονται στις εικόνες με σκοπό την απόδοση χαρακτηριστικών, όπως γραμμές, άκρες και μοτίβα. Έπειτα, τα χαρακτηριστικά αυτά χρησιμοποιούνται για την κατηγοριοποίηση ή την ανίχνευση αντικειμένων στις εικόνες.



Σχήμα 1: Αρχιτεκτονική δικτύου CNN[15]

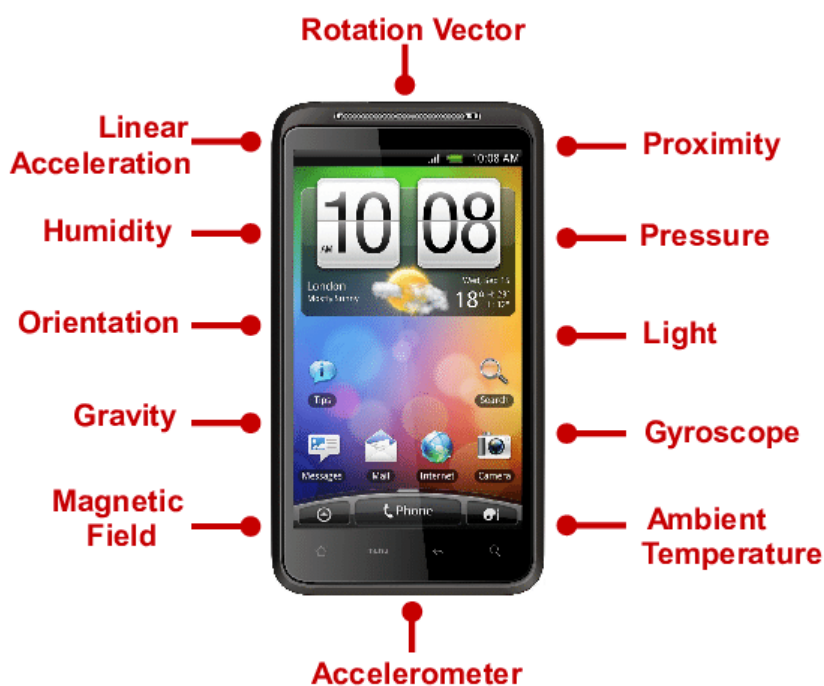
2.3 Αισθητήρες

Οι αισθητήρες είναι ηλεκτρονικές συσκευές που ανιχνεύουν και μετατρέπουν φυσικά σήματα από το περιβάλλον σε ηλεκτρονικά σήματα που μπορούν να αναγνωστούν από μια μηχανή. Η χρήση τους διευρύνεται σε μια ποικιλία εφαρμογών, όπως στην επιστήμη των υπολογιστών, την ιατρική, τον βιομηχανικό έλεγχο και τη ρομποτική.

Οι αισθητήρες λειτουργούν με τη χρήση διαφόρων τεχνολογιών, όπως αισθητήρες πίεσης, επιτάχυνσης, γυροσκοπίων, μαγνητικών πεδίων, θερμοκρασίας και φωτεινότητας, και επικεντρώνονται στη μέτρηση της παρουσίας και της έντασης των σημάτων αυτών και στη μετατροπή τους σε ψηφιακά σήματα, τα οποία καταφέρνουν να αναγνωστούν από μια μηχανή.

Στην παρούσα εργασία, για παράδειγμα, χρησιμοποιείται το Sussex Locomotion Dataset, το οποίο περιλαμβάνει καταγραφές επιταχύνσεων και γυροσκοπικών μετρήσεων από αισθητήρες που τοποθετήθηκαν σε κινητά τηλέφωνα. Οι αισθητήρες στα κινητά τηλέφωνα καταγράφουν τις επιταχύνσεις και τις περιστροφές του τηλεφώνου κατά τη διάρκεια χρήσης του. Με αυτό τον τρόπο, είναι δυνατή η συλλογή δεδομένων για τις δραστηριότητες των χρηστών, όπως περπάτημα, τρέξιμο, ποδηλασία, οδήγηση, κτλ.

Η διαδικασία μετατροπής των σημάτων από τους αισθητήρες σε ψηφιακή μορφή περιλαμβάνει συνήθως μια σειρά επεξεργασιών. Αρχικά, οι αναλογικοί αισθητήρες παράγουν ένα αναλογικό σήμα, το οποίο αποτελεί μια αναλογική αναπαράσταση του σήματος που ανιχνεύουν. Έπειτα, αυτό το αναλογικό σήμα μετατρέπεται σε ψηφιακό μέσω ενός αναλογοψηφιακού μετατροπέα (ADC). Τέλος, το ψηφιακό σήμα επεξεργάζεται από μια μηχανή που μπορεί να το αποθηκεύσει, να το επεξεργαστεί και να το αναλύσει.



Σχήμα 2: Αισθητήρες σε κινητό[20]

2.3.1 Επιταχυνσόμετρο

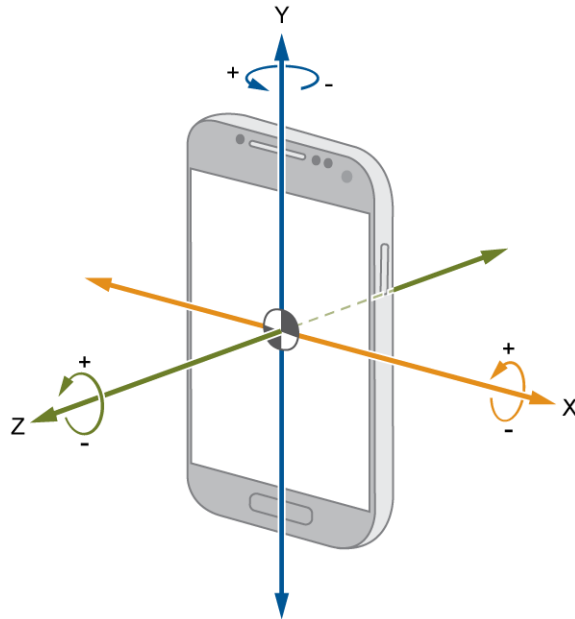
Το επιταχυνσόμετρο[17] είναι ένας αισθητήρας που μετρά την επιτάχυνση σε μια συγκεκριμένη διεύθυνση. Συνήθως, τα επιταχυνσόμετρα χρησιμοποιούνται για να μετρήσουν την επιτάχυνση της βαρύτητας σε ένα σταθερό άξονα, αλλά μπορούν επίσης να μετρήσουν την επιτάχυνση και σε άλλες διευθύνσεις, όπως οι επιταχύνσεις που προκύπτουν από κινήσεις σε επίπεδο έδαφος.

2.3.2 Μαγνητόμετρο

Το μαγνητόμετρο[4] είναι ένας αισθητήρας που μετρά το μαγνητικό πεδίο σε έναν συγκεκριμένο χώρο. Αυτό το μαγνητικό πεδίο μπορεί να προέρχεται από μαγνήτες, ηλεκτρομαγνήτες ή ακόμη και από τον ανθρώπινο εγκέφαλο. Το μαγνητόμετρο λειτουργεί σε συνδυασμό με άλλους αισθητήρες, όπως το επιταχυνσόμετρο και ο γυροσκόπος, ώστε να παράγει μια ολοκληρωμένη εικόνα της κίνησης της συσκευής και του χρήστη που τη χειρίζεται. Συνολικά, οι αισθητήρες αυτοί συνεργάζονται για να παράγουν ένα πλήρες προφίλ της κίνησης του χρήστη, καθώς και των χαρακτηριστικών της κίνησης αυτής, όπως η ταχύτητα και η επιτάχυνση. Η συλλογή και η ανάλυση των δεδομένων αυτών μπορεί να χρησιμοποιηθεί για τη βελτίωση της αντίληψης των ανθρώπων για την κίνηση και την ανάπτυξη προηγμένων συστημάτων αυτόματης αναγνώρισης της κίνησης, τα οποία έχουν τη δυνατότητα ύπαρξης εφαρμογών σε πολλούς τομείς, όπως η ιατρική, η ρομποτική, η ασφάλεια και η ψυχαγωγία.

2.3.3 Γυροσκόπιο

Το γυροσκόπιο[5] αποτελεί έναν αισθητήρα που χρησιμοποιείται για την καταγραφή της γωνιακής ταχύτητας περιστροφής του σώματος ενός ανθρώπου καθώς κινείται. Μπορεί να τοποθετηθεί σε μια συσκευή που τοποθετείται για παράδειγμα στον αστράγαλο, και να καταγράφει τις αλλαγές της γωνίας κατά την κίνηση του ατόμου. Η χρήση του γυροσκοπίου συνδυάζεται συνήθως με άλλους αισθητήρες, όπως το επιταχυνσόμετρο και το μαγνητόμετρο, με σκοπό να καταγράψουν πλήρη εικόνα της κίνησης του ανθρώπου και της δραστηριότητάς του.



Σχήμα 3: Πως είναι στους σένσορες τα X,Y,Z[18]

2.4 Βιβλιογραφική επισκόπηση

Το 2017 δημιουργήθηκε το SHL Dataset[24] από τρεις χρήστες, οι οποίοι χρησιμοποίησαν από ένα κινητό, μέσω του οποίου καταγράφονταν συγκεκριμένες καθημερινές κινήσεις που έκαναν. Εξαιτίας της πληθώρας των δεδομένων που καταγράφηκαν, το Dataset θεωρείται ότι είναι το μεγαλύτερο ευρέως προσβάσιμο dataset στο επιστημονικό πεδίο. Πραγματοποιήθηκαν τρία πειράματα σε τρεις διαδοχικές χρονιές (2018 – 2020) και σε καθένα από αυτά χρησιμοποιήθηκαν train data και test data. Στο πρώτο πείραμα εστίασαν στα χρονικά περιθώρια μεταξύ των δραστηριοτήτων, χρησιμοποιώντας το κινητό στη θέση της τσέπης του χρήστη 1. Τα αποτελέσματα εξήλθαν από 19 καταγραφές που συλλέχθηκαν από 17 ομάδες στο διάστημα ενάμισι μηνός. Στο δεύτερο πείραμα πραγματοποιήθηκε η θέση του κινητού (4 διαφορετικές θέσεις) και τα δεδομένα προήλθαν ξανά από τον χρήστη 1. Το χρονικό διάστημα για την εξαγωγή των αποτελεσμάτων ήταν ενάμισης μήνας και συγκεντρώθηκαν 14 καταγραφές από 14 ομάδες. Στο τρίτο πείραμα, τα δεδομένα προήλθαν κυρίως από τον χρήστη 1, αλλά και από τους άλλους δύο χρήστες. Τα train data χρησιμοποιήθηκαν από τον χρήστη 1, ενώ τα test data από τους χρήστες 2 και 3, διότι δεν μπορούσαν να συμμετάσχουν σε όλες τις δραστηριότητες. Το διάστημα αυτή τη φορά ήταν δύομισι μήνες και εξάχθηκαν 15 αποτελέσματα από 15 ομάδες. Τα αποτελέσματα εξάχθηκαν από τα test data, τα οποία προβλέπουν το F1 score για όλα τα πειράματα. Στο πείραμα του 2018 το F1 score παρουσίασε διάστημα από 53,2 έως 93,9%, στο πείραμα του 2019 είχε διάστημα 31,5 έως 78,4%, ενώ στο πείραμα του 2020 ήταν από 17,8 έως 88,5%. Υπήρξαν ML και DL προσεγγίσεις, όπου τα νευρωνικά δίκτυα, παρόλο που ήταν πιο ‘άγνωστα’ στους συμμετέχοντες, αποτύπωναν καλύτερα αποτελέσματα σε αντίθεση με τις απλές μεθόδους της μηχανικής μάθησης. Όπως φαίνεται και στην παρούσα εργασία χρησιμοποιήθηκε βαθιά μηχανική αφού είναι πιο αποδοτική. Βέβαια, η μηχανική μάθηση είχε και αυτή καλύτερα αποτελέσματα στο δεύτερο πείραμα αλλά στο τρίτο η διαφορά ήταν αρκετά εμφανής – 88.5% στη βαθιά νευρωνική μάθηση με 77.9% στην μηχανική μάθηση. Βλέπουμε μία διαφορά σχεδόν 10% λιγότερη οπότε το συμπέρασμα είναι πως όσο αναπτύσσεται η DL τόσο περισσότερο θα χρησιμοποιείται.

3 Αλγόριθμοι μέτρησης της κίνησης

3.1 Περιγραφή του Συνόλου Δεδομένων και της Προεπεξεργασίας δεδομένων

3.1.1 Σύνολο Δεδομένων

Το Sussex Huawei Locomotion (SHL)[11][23] είναι ένα σύνολο δεδομένων που χρησιμοποιείται στην παροχή πληροφοριών σχετικά με τη φυσική δραστηριότητα των ανθρώπων, οι οποίες έχουν καταγραφεί με τη βοήθεια αισθητήρων τοποθετημένων σε διάφορα σημεία του σώματος των ατόμων. Το σύνολο αυτό περιλαμβάνει ένα εύρος δραστηριοτήτων και μετακινήσεων, όπως περπάτημα, τρέξιμο, ποδηλασία, οδήγηση και άλλες καθημερινές δραστηριότητες.

Το dataset βρίσκει εφαρμογή σε ποικίλους τομείς. Ειδικότερα, στο χώρο της υγείας, η χρήση του αφορά την παροχή στοιχείων σχετικά με το επίπεδο φυσικής δραστηριότητας του χρήστη, ενώ στο χώρο της φορητής τεχνολογίας, τα δεδομένα του dataset επιτυγχάνουν τη βελτιστοποίηση και εξατομίκευση των φορητών συσκευών με βάση τις ανάγκες του χρήστη.

Για τη συλλογή των δεδομένων, τοποθετούνται συνολικά 4 κινητά τηλέφωνα σε 4 διαφορετικά σημεία του σώματος. Το πρώτο, που είναι και το «βασικό», κρατείται στο χέρι, ενώ το δεύτερο, τρίτο και τέταρτο τοποθετούνται στο στήθος, σε εμπρόσθια τσέπη στο ύψος των γοφών, και μέσα σε τσάντα αντίστοιχα. Τα δεδομένα αυτά καταγράφονται στο SHL και αφορούν πληροφορίες όπως ο χρόνος έναρξης και λήξης κάθε δραστηριότητας, η ταχύτητα και η επιτάχυνση των κινήσεων, η απόσταση που διένυσε ο χρήστης κατά τη διάρκεια εκτέλεσης των δραστηριοτήτων, καθώς επίσης και πληροφορίες σχετικά με τα χαρακτηριστικά της φυσιολογίας του ατόμου, όπως ο καρδιακός ρυθμός. Κάθε κινητό τηλέφωνο έχει εγκατεστημένη μια εφαρμογή, η οποία περιλαμβάνει τα δεδομένα του dataset, και είναι απαραίτητη για τη συλλογή και την καταγραφή των πληροφοριών από τους χρήστες. Τα δεδομένα αποθηκεύονται σε μορφή .txt και διαχωρίζονται σε 9 κατηγορίες, το σύνολο των οποίων αποθηκεύεται συνολικά 4 φορές, μία για κάθε κινητό τηλέφωνο. Οι κατηγορίες αυτές είναι:

- Ambient
- API
- Battery
- Cells
- DeprCells
- GPS
- Location

- Motion
- WiFi

Επιπλέον, υπάρχουν κι άλλα αρχεία που χρησιμοποιούνται στη συλλογή και κατηγοριοποίηση των δεδομένων, κυριότερα από τα οποία είναι τα αρχεία Label. Το στοιχείο που χαρακτηρίζει τα Label αρχεία είναι οι πληροφορίες σχετικά με τη δραστηριότητα του χρήστη.

Column	What																																						
1	Time [ms]																																						
2	Coarse label: Null=0, Still=1, Walking=2, Run=3, Bike=4, Car=5, Bus=6, Train=7, Subway=8																																						
3	Fine label: <table> <tr><td>Null=</td><td>0</td></tr> <tr><td>Still;Stand;Outside=</td><td>1</td></tr> <tr><td>Still;Stand;Inside=</td><td>2</td></tr> <tr><td>Still;Sit;Outside=</td><td>3</td></tr> <tr><td>Still;Sit;Inside=</td><td>4</td></tr> <tr><td>Walking;Outside=</td><td>5</td></tr> <tr><td>Walking;Inside=</td><td>6</td></tr> <tr><td>Run=</td><td>7</td></tr> <tr><td>Bike=</td><td>8</td></tr> <tr><td>Car;Driver=</td><td>9</td></tr> <tr><td>Car;Passenger=</td><td>10</td></tr> <tr><td>Bus;Stand=</td><td>11</td></tr> <tr><td>Bus;Sit=</td><td>12</td></tr> <tr><td>Bus;Up;Stand=</td><td>13</td></tr> <tr><td>Bus;Up;Sit=</td><td>14</td></tr> <tr><td>Train;Stand=</td><td>15</td></tr> <tr><td>Train;Sit=</td><td>16</td></tr> <tr><td>Subway;Stand=</td><td>17</td></tr> <tr><td>Subway;Sit=</td><td>18</td></tr> </table>	Null=	0	Still;Stand;Outside=	1	Still;Stand;Inside=	2	Still;Sit;Outside=	3	Still;Sit;Inside=	4	Walking;Outside=	5	Walking;Inside=	6	Run=	7	Bike=	8	Car;Driver=	9	Car;Passenger=	10	Bus;Stand=	11	Bus;Sit=	12	Bus;Up;Stand=	13	Bus;Up;Sit=	14	Train;Stand=	15	Train;Sit=	16	Subway;Stand=	17	Subway;Sit=	18
Null=	0																																						
Still;Stand;Outside=	1																																						
Still;Stand;Inside=	2																																						
Still;Sit;Outside=	3																																						
Still;Sit;Inside=	4																																						
Walking;Outside=	5																																						
Walking;Inside=	6																																						
Run=	7																																						
Bike=	8																																						
Car;Driver=	9																																						
Car;Passenger=	10																																						
Bus;Stand=	11																																						
Bus;Sit=	12																																						
Bus;Up;Stand=	13																																						
Bus;Up;Sit=	14																																						
Train;Stand=	15																																						
Train;Sit=	16																																						
Subway;Stand=	17																																						
Subway;Sit=	18																																						
4	Road label: City=1, Motorway=2, Countryside=3, Dirt road=4, Null=0																																						
5	Traffic label: Heavy traffic=1, null=0																																						
6	Tunnels label: Tunnel=1, null=0																																						
7	Social label: Social=1, null=0																																						
8	Food label: Eating=1, Drinking=2, Both=3, null=4																																						

Σχήμα 4: Label.txt μορφή

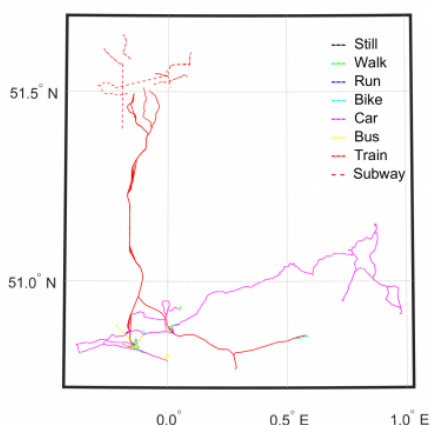
3.1.2 Προεπεξεργασία δεδομένων

Για να χρησιμοποιηθούν τα δεδομένα, είναι απαραίτητη η διαμόρφωσή τους. Χρησιμοποιούνται 2 αρχεία στα οποία θα γίνει μια μικρή επεξεργασία, ώστε να αφαιρεθούν οι στήλες που δεν χρειάζονται σε κανένα από αυτά και να συγχωνευθούν τα δύο αρχεία σε ένα. Η συγχώνευση γίνεται κατά πλάτος και δημιουργείται ένας πίνακας με στήλες. Αφού ολοκληρωθεί αυτή η διαδικασία, γίνεται διαγραφή των στηλών με τα NaN δεδομένα εφόσον δεν χρειάζονται. Έπειτα, γίνεται μια κανονικοποίηση των δεδομένων, δηλαδή η συνάρτηση επιστρέφει τις τιμές των στηλών που δόθηκαν να μετακινούνται στο εύρος (-1,1). Αυτό έχει ως στόχο την ευκολότερη επεξεργασία τους. Το τελευταίο βήμα είναι η λήψη του νέου αρχείου με τα δεδομένα συγχωνευμένα και ο διαχωρισμός του σε 2 αρχεία με τα καινούρια δεδομένα. Εν ολίγοις, η προεπεξεργασία και ο διαχωρισμός είναι απαραίτητος για να γίνει πιο εύκολη και να κυλίσει ομαλά η εκπαίδευση των μοντέλων.

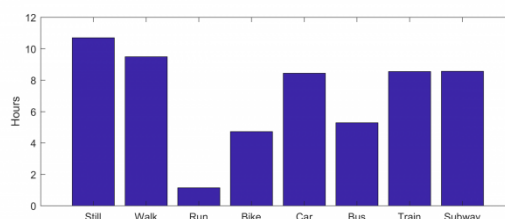
3.2 Περιγραφή του σετ εκπαίδευσης, δοκιμής και επικύρωσης

Τα σύνολα εκπαίδευσης, δοκιμής και επικύρωσης είναι τρία διακριτά υποσύνολα ενός μεγαλύτερου συνόλου δεδομένων που χρησιμοποιούνται για εκπαίδευση, αξιολόγηση και συντονισμό μοντέλων μηχανικής μάθησης. Το εκπαιδευτικό σύνολο χρησιμοποιείται για την εκπαίδευση ενός μοντέλου, με στόχο την εκμάθηση των σχέσεων στα δεδομένα. Ο αλγόριθμος χρησιμοποιεί τα χαρακτηριστικά και τις ετικέτες του συνόλου εκπαίδευσης για να ενημερώσει τις παραμέτρους του και να κάνει προβλέψεις.

Το σύνολο δοκιμών χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός εκπαιδευμένου μοντέλου, χρησιμοποιώντας το για να κάνει προβλέψεις σε δεδομένα που δεν έχει δει ποτέ πριν. Παρέχει, επίσης, ένα μέτρο για το πόσο καλά το μοντέλο γενικεύεται σε νέα, άγνωστα δεδομένα. Το σύνολο επικύρωσης χρησιμοποιείται για τον συντονισμό των παραμέτρων του μοντέλου, παράμετροι που ελέγχουν την διαδικασία εκμάθησης. Επιπλέον, χρησιμοποιείται για την σύγκριση διαφορετικών μοντέλων και για τον προσδιορισμό του καλύτερου συνόλου παραμέτρων για μια διεργασία.



Σχήμα 5: Dataset preview[11][23]



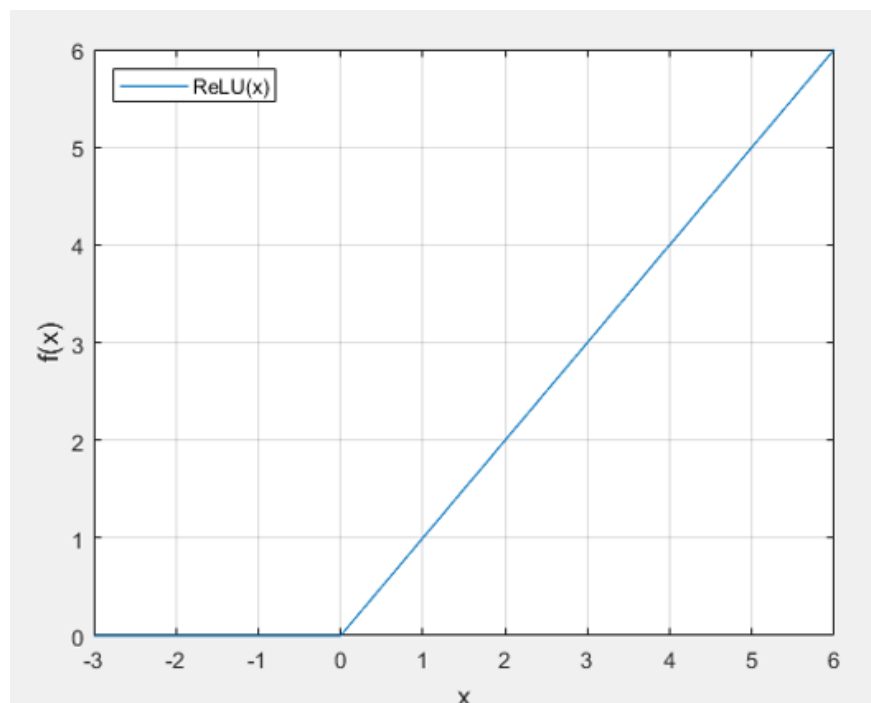
Σχήμα 6: Dataset plot[11][23]

3.3 Συναρτήσεις Ενεργοποίησης

3.3.1 Ανορθωμένη Γραμμική Μονάδα – ReLU

Η ReLU[6] είναι μια ευρέως χρησιμοποιημένη συνάρτηση ενεργοποίησης στην βαθιά μάθηση και στα συνελκτικά νευρωνικά δίκτυα. Είναι μια μισοδιορθωμένη συνάρτηση, που σημαίνει ότι η $f(z)$ είναι μηδέν, όταν το z είναι μικρότερο από 0 και ίση με z όταν το z είναι πάνω ή ίσο με 0. Το εύρος της ReLU είναι $[0, \infty)$. Τόσο η συνάρτηση όσο και η παράγωγός της είναι μονοτονικές. Ωστόσο, ένα μειονέκτημα της ReLU είναι ότι οι αρνητικές εισροές μηδενίζονται αμέσως, μειώνοντας την ικανότητα του μοντέλου να ταιριάζει σωστά ή να εκπαιδεύει τα δεδομένα, καθώς επηρεάζει την αντιστοίχιση των αρνητικών τιμών στο γράφημα.

Η ReLU συνάρτηση ορίζεται με την παρακάτω εξίσωση : $f(x) = \max(0, x)$

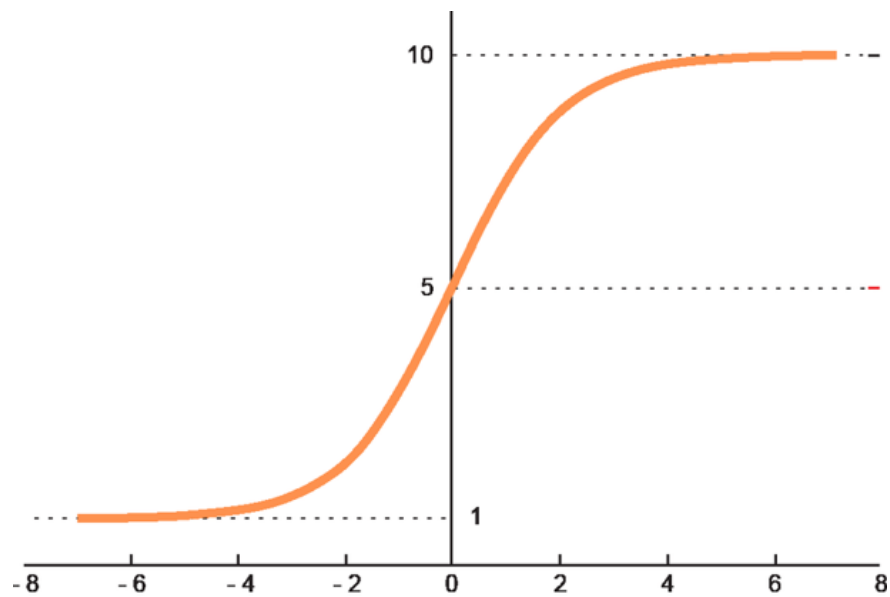


Σχήμα 7: ReLU[27]

3.3.2 Συνάρτηση SoftMax

Η συνάρτηση SoftMax[26] μετατρέπει ένα σύνολο αριθμών σε κατανομή πιθανότητας όπου η πιθανότητα κάθε αριθμού είναι ανάλογη με το μέγεθος του στο αρχικό διάνυσμα. Χρησιμοποιείται ευρέως ως συνάρτηση ενεργοποίησης σε νευρωνικά δίκτυα για ταξινόμηση πολλαπλών κλάσεων. Το δίκτυο εξάγει N τιμές όπου αντιστοιχούν σε κάθε κλάση και η SoftMax τις κάνει κανονικοποίηση σε πιθανότητες που αθροίζονται σε 1. Αυτές οι πιθανότητες αντιπροσωπεύουν την πιθανότητα κάθε τιμής εξόδου να ανήκει σε μια συγκεκριμένη κλάση.

Η SoftMax συνάρτηση ορίζεται με την παρακάτω εξίσωση : $f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$



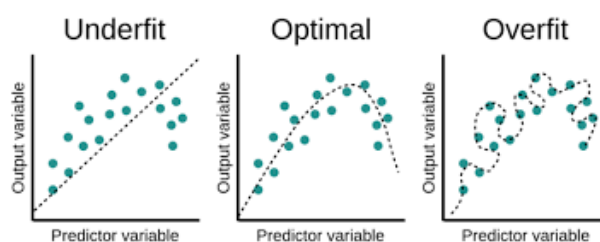
Σχήμα 8: Softmax[7]

3.4 Τύποι υπερ-παραμέτρων

Οι υπέρ-παράμετροι στα νευρωνικά δίκτυα είναι βασικές παράμετροι που δεν μαθαίνονται κατά τη διάρκεια της εκπαίδευσης, αλλά πρέπει να ρυθμιστούν πριν από την εκπαίδευση. Είναι κρίσιμες παράμετροι που επηρεάζουν την απόδοση και τη συμπεριφορά του νευρωνικού δικτύου κατά τη διάρκεια της εκπαίδευσης και μπορούν να έχουν σημαντικό αντίκτυπο στην τελική απόδοση του μοντέλου. Ένα νευρωνικό δίκτυο είναι ένα πολύπλοκο σύστημα με πολλά διασυνδεδεμένα στοιχεία. Επομένως, η επιλογή των σωστών τιμών για τις υπέρ-παραμέτρους είναι κρίσιμη για τον προσδιορισμό της επιτυχίας ενός μοντέλου νευρωνικών δικτύων. Υπάρχουν αρκετές υπέρ-παραμέτροι που πρέπει να ρυθμιστούν πριν από την εκπαίδευση του νευρωνικού δικτύου. Μία από τις πιο κρίσιμες υπέρ-παραμέτρους είναι ο ρυθμός εκμάθησης. Αυτή η παράμετρος ελέγχει το μέγεθος των βημάτων που γίνονται κατά τη βελτιστοποίηση και επηρεάζει την ταχύτητα και την ποιότητα της μάθησης. Επίσης, ένας μεγάλος ρυθμός μάθησης μπορεί να οδηγήσει σε γρήγορη σύγκλιση, αλλά ο αλγόριθμος μπορεί να υπερβεί τη βέλτιστη λύση, αντιθέτως ένας μικρός ρυθμός εκμάθησης μπορεί να χρειαστεί πολύ χρόνο για να συγκλίνει.

Η επιλογή των συναρτήσεων ενεργοποίησης σε κάθε επίπεδο αποτελεί μια από τις υπερ-παραμέτρους που επηρεάζουν την μη γραμμικότητα του μοντέλου. Αυτή η επιλογή έχει σημαντικό αντίκτυπο στην απόδοση του μοντέλου. Έτσι, η σωστή επιλογή της συνάρτησης ενεργοποίησης μπορεί να κάνει την διαφορά μεταξύ ενός νευρωνικού δικτύου που έχει ικανοποιητική απόδοση και ενός δικτύου που έχει ελλιπή. Συνεπώς, η επιλογή της κατάλληλης συνάρτησης ενεργοποίησης αποτελεί σημαντικό βήμα στον σχεδιασμό και την εκπαίδευση ενός νευρωνικού δικτύου. Κάθε συνάρτηση ενεργοποίησης προσφέρει διαφορετική ικανότητα για το μοντέλο να αντιλαμβάνεται και να προσαρμόζεται σε πολύπλοκες διασυνδέσεις και μοτίβα στα δεδομένα.

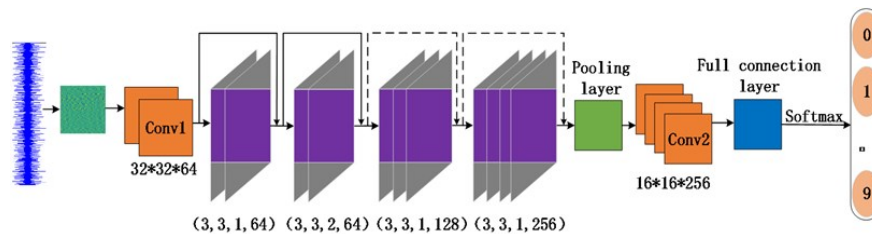
Μαζί με τις υπερ-παραμέτρους υπάρχει και μια τεχνική, τακτοποίηση (regularization)[1], η οποία χρησιμοποιείται για την αποφυγή υπερβολικής προσαρμογής ενός μοντέλου. Έχει 2 μορφές και είναι είτε L1 είτε L2. Αυτές οι τεχνικές τακτοποίησης εφαρμόζονται με τον καθορισμό συγκεκριμένων υπέρ-παραμέτρων όπως, η ισχύς της ποινής τακτοποίησης ή η πιθανότητα διακοπής. Η τακτοποίηση είναι απαραίτητη για την αποφυγή υπερβολικής προσαρμογής, η οποία μπορεί να συμβεί όταν ένα μοντέλο είναι περίπλοκο και ταιριάζει πολύ στα δεδομένα εκπαίδευσης. Το μέγεθος παρτίδας (batch size) είναι μια υπέρ-παράμετρος που ελέγχει τον αριθμό των δειγμάτων που υποβάλλονται σε επεξεργασία μαζί κατά τη διάρκεια κάθε επανάληψης της εκπαίδευσης. Συνοψίζοντας, οι υπέρ-παράμετροι είναι καθοριστικές για τον προσδιορισμό της απόδοσης και της συμπεριφοράς του νευρωνικού δικτύου κατά τη διάρκεια της εκπαίδευσης.



Σχήμα 9: Regularization[1]

3.5 Adam Optimizer

Ο αλγόριθμος Adam (Adaptive Moment Estimation)[2] είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται κυρίως για την εκπαίδευση νευρωνικών δικτύων σε προβλήματα μηχανικής μάθησης. Σκοπός της λειτουργίας του αλγορίθμου είναι η όσο το δυνατόν πιο αποτελεσματική και γρήγορη εκμάθηση ενός μοντέλου νευρωνικού δικτύου.



Σχήμα 10: Αρχιτεκτονική Adam optimizer[25]

3.6 Μετρικές

Πληροφορίες σχετικά με την απόδοση του μοντέλου δίνονται από τέσσερις μετρικές[22], οι οποίες είναι χρήσιμες ανάλογα με τον συγκεκριμένο στόχο και τα χαρακτηριστικά του προβλήματος κατηγοριοποίησης.

3.6.1 Ευστοχία (Accuracy)

Η ευστοχία είναι μια μετρική που μας λέει πόσο συχνά το μοντέλο μας προβλέπει σωστά την κλάση ενός δείγματος. Επίσης, είναι ένα χρήσιμο μέτρο για προβλήματα ισορροπημένων κατηγοριών, όπου οι κλάσεις έχουν παρόμοιο αριθμό δειγμάτων.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives}$$

3.6.2 Ακρίβεια (Precision)

Σκοπός της ακρίβειας είναι να επιβεβαιώσει πως οι προβλέψεις για μια συγκεκριμένη κατηγορία είναι πραγματικά σωστές. Η ακρίβεια αξιολογεί πόσο απόλυτα ακριβείς είναι οι θετικές προβλέψεις σε σχέση με τον συνολικό αριθμό των προβλέψεων.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

3.6.3 Ανάκληση (Recall)

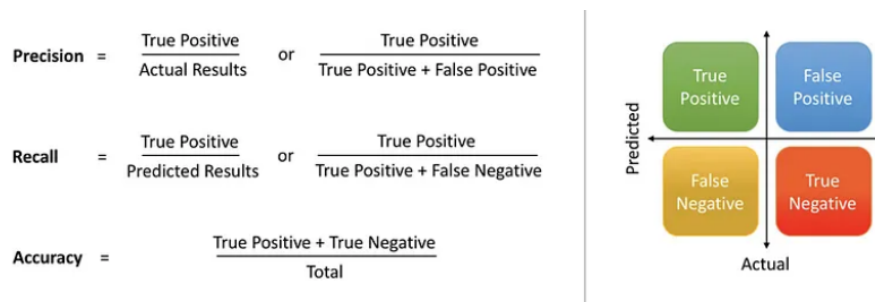
Η ανάκληση χρησιμεύει στη μέτρηση του αριθμού των θετικών περιπτώσεων που προέβλεψε σωστά ο ταξινομητής, σε όλες τις θετικές περιπτώσεις στα δεδομένα.

$$Recall = \frac{TruePositives}{TruePositives+FalsePositives}$$

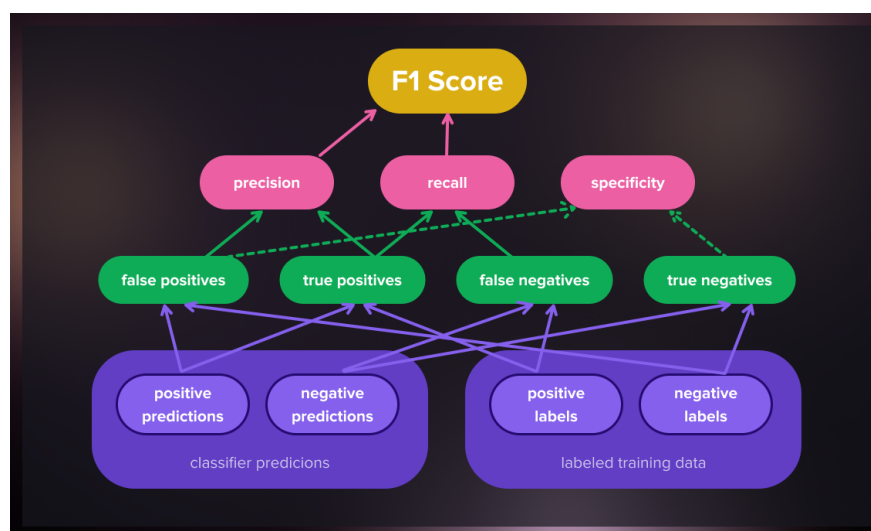
3.6.4 F1 score

Το F1 score[14] είναι σημαντικό σε περιπτώσεις ανισορροπίας μεταξύ των κλάσεων και είναι επιθυμητή η αξιολόγηση του μοντέλου σε κάθε κλάση. Το F1 score είναι ένας συνδυασμός της ακρίβειας και της ανάκλησης.

$$F1Score = 2 * \frac{Precision*Recall}{Precision+Recall}$$



Σχήμα 11: Ακρίβεια και ανάκληση[21]



Σχήμα 12: F1 score[14]

4 Documentation του Κώδικα

4.1 Preprocessing

Με την χρήση του preprocessing με την χρήση μια μεταβλητής διαβάζονται τα δεδομένα, είτε για πολλούς χρήστες είτε για έναν χρήστη. Έπειτα, επιλέγεται η θέση του κινητού που θα χρησιμοποιηθεί και ξεκινάει η προ-επεξεργασία των δεδομένων. Κατά την διάρκεια της προ-επεξεργασίας κρατιούνται μόνο οι στήλες που θα χρησιμοποιηθούν και γίνεται μια κανονικοποίηση στα δεδομένα των σενσορών από το -1 έως το 1 πριν αποθηκευτούν.

```
1 # Function for normalizing data in each column but not the label column.
2 def normalize(df):
3     for i in range(3):
4         df[i] = preprocessing.MinMaxScaler(feature_range=(-1, 1)).fit_transform(
5             np.array(df[i]).reshape(-1, 1))
6     return df
7
8 if __name__ == '__main__':
9     # Load motion data and labels.
10    print('Loading motion data\n')
11    data_df = pd.read_csv('./Hand_Motion.txt', delim_whitespace=True, header=
12        None)
13
14    print('Loading label data\n')
15    label_df = pd.read_csv('./Label.txt', delim_whitespace=True, header=None)
16
17    # Drop unnecessary columns.
18    label_df = label_df.drop(label_df.iloc[:, 2:8], axis=1)
19    label_df = label_df.drop(label_df.columns[[0]], axis=1)
20
21    data_df = data_df.drop(data_df.iloc[:, 4:23], axis=1)
22    data_df = data_df.drop(data_df.columns[[0]], axis=1)
23
24    # Concatenate the dataframes.
25    data_df = pd.concat([data_df, label_df], ignore_index=True, sort=False,
26        axis=1, join='inner')
27
28    # Drop NaN rows.
29    data_df = data_df.dropna()
30
31    # Normalize data between -1 and 1.
32    data_df = normalize(data_df)
33
34    # Write the final dataframe.
35    data_df.to_csv('Preprocessed_Data.csv', index=False, sep=';', header=False
36        )
```

Source code 1: Preprocessing code

4.2 Data Handler

Διαβάζει τα αρχεία που αποθηκεύτηκαν πριν και με βάση την μεταβλητή για το αν πρέπει να διαβάσει για πολλαπλούς χρήστες ξεκινάει να αφαιρεί στήλες με σενσορές που δεν θα χρησιμοποιηθούν. Τέλος, αποθηκεύει τα δεδομένα από τους σένσορες και σε ξεχωριστό αρχείο τις ετικέτες[12].

```
1 if __name__ == '__main__':
2     # Load motion data and labels.
3     print('Loading motion data\n')
4     df = pd.read_csv('./Preprocessed_Data.csv', sep=';', header=None)
5
6     df_x = df.drop(df.columns[[3]], axis=1)
7
8     # Write the final dataframe.
9     df_x.to_csv('data.csv', index=False, sep=';', header=False)
10
11    df_y = df.drop(df.columns[[0, 1, 2]], axis=1)
12
13    # Write the final dataframe.
14    df_y.to_csv('labels.csv', index=False, sep=';', header=False)
```

Source code 2: Data Handler code

4.3 CNN – CNN + LSTM - CNN + GRU

Διαβάζοντας τα προ-επεξεργασμένα δεδομένα, τις ετικέτες και αφού γίνουν οι κατάλληλες μετατροπές στα δεδομένα. Πρώτα στην περίπτωση που τα δεδομένα αφορούν έναν χρήστη, χωρίζονται σε train και test. Αν υπάρχουν δεδομένα για πολλαπλούς χρήστες τότε, πρέπει να καταχωρηθεί ποιος χρήστης θα μείνει εκτός για δοκιμές leave-one-out. Έπειτα, το αρχείο με τις ετικέτες θα γίνει της μορφής one-hot encoding[9]. Τέλος, δημιουργείται το νευρωνικό δίκτυο, οποιασδήποτε αρχιτεκτονικής είτε CNN είτε CNN+LSTM είτε CNN+GRU, για να γίνει η εκπαίδευση του μοντέλου, να αποθηκευτεί το μοντέλο και να αποθηκευτούν οι εικόνες με τις μετρικές εκπαίδευσης.

```

1 if __name__ == '__main__':
2     # Read the data.
3     x = pd.read_csv('./data.csv', sep=';', header=None).to_numpy()
4     y = pd.read_csv('./labels.csv', sep=';', header=None).to_numpy()
5
6     # Get number of classes.
7     classes = np.unique(y).size
8
9     # Change labels to a one-hot encoding format.
10    y = keras.utils.to_categorical(y - 1, num_classes=classes)
11
12    # Split data to train test with the corresponding label.
13    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
14
15    # Create the model based on the VGG-16
16    model = Sequential()
17    model.add(Conv1D(input_shape=(3, 1), filters=64, kernel_size=3, padding="
        same", activation="relu",))
18    model.add(Conv1D(filters=32, kernel_size=3, padding="same", activation="
        relu"))
19    model.add(MaxPool1D(pool_size=2, strides=2))
20    model.add(Conv1D(filters=64, kernel_size=3, padding="same", activation="
        relu"))
21    model.add(Conv1D(filters=64, kernel_size=3, padding="same", activation="
        relu"))
22    model.add(MaxPool1D(pool_size=1, strides=1))
23    model.add(Flatten())
24    model.add(Dense(units=512, activation="relu"))
25    model.add(Dense(units=256, activation="relu"))
26    model.add(Dense(units=classes, activation="softmax"))
27
28    # Change the optimization.
29    opt = Adam(learning_rate=0.001)
30    model.compile(optimizer=opt, loss=keras.losses.categorical_crossentropy,
        metrics=['accuracy'])
31
32    # Show the summary of the model.
33    model.summary()
34
35    # Set the callbacks.
36    checkpoint = ModelCheckpoint("vgg16_1.h5", monitor='val_accuracy', verbose
        =1, save_best_only=True, save_weights_only=False, mode='auto', save_freq
        =1)
37    early = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=20,
        verbose=1, mode='auto')
38
39    # Train the model.
40    hist = model.fit(x_train, y_train, epochs=100, batch_size=10000, callbacks
        =[checkpoint, early], verbose=1, validation_split=0.2)
41

```



```

42 # Plot and save the Model accuracy.
43 plt.plot(hist.history["accuracy"])
44 plt.plot(hist.history['val_accuracy'])
45 plt.plot(hist.history['loss'])
46 plt.plot(hist.history['val_loss'])
47 plt.title("CNN-VGG Model Accuracy")
48 plt.ylabel("Accuracy")
49 plt.xlabel("Epoch")
50 plt.legend(["Accuracy", "Validation Accuracy", "loss", "Validation Loss"])
51 plt.savefig('Model_Accuracy_CNN_VGG.png')

```

Source code 3: CNN

Ίδιος κώδικας όπου προστίθεται το μοντέλου LSTM.

```

1 # Define CNN model.
2 cnn = Sequential()
3 cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu', padding='same', input_shape=(3, 1)))
4 cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu', padding='same', input_shape=(3, 1)))
5 cnn.add(Dropout(0.5))
6 cnn.add(MaxPooling1D(pool_size=1))
7 cnn.add(Flatten())
8 # Define LSTM model.
9 model = Sequential()
10 model.add(TimeDistributed(cnn))
11 model.add(LSTM(units=100))
12 model.add(Dropout(0.5))
13 model.add(Dense(units=20, activation='relu'))
14 model.add(Dense(classes, activation='softmax'))

```

Source code 4: CNN-LSTM

Παρόμοια, απλά προστίθεται το μοντέλο GRU.

```

1 # Define CNN model.
2 model = Sequential()
3 model.add(Conv1D(64, 3, activation='relu', input_shape=input_shape))
4 model.add(SpatialDropout1D(0.2))
5 model.add(MaxPooling1D(2))
6 model.add(GRU(64))
7 model.add(Dropout(0.1))
8 model.add(Dense(classes, activation='softmax'))
9 model.summary()

```

Source code 5: CNN-GTU

4.4 Load Model

Για να γίνουν οι δοκιμές και να εξαχθούν αποτελέσματα χρησιμοποιούνται τα μοντέλα που αποθηκεύτηκαν. Διαβάζοντας τα κατάλληλα αρχεία, όπως στην περίπτωση του leave-one-out να υπάρχουν τα δεδομένα για τον χρήστη που δεν εκπαιδεύτηκε το μοντέλο πάνω. Αφού γίνει το φόρτωμα του μοντέλου, τρέχει η αξιολόγηση του μοντέλου με τις ίδιες μετρικές που υπήρχαν και κατά την εκπαίδευση για να υπάρχει παρόμοιο μέτρο σύγκρισης.

```
1 def load_data():
2     # Read the data.
3     x = pd.read_csv('./data.csv', sep=';', header=None).to_numpy()
4     y = pd.read_csv('./labels.csv', sep=';', header=None).to_numpy()
5
6     # Get number of classes and input shape.
7     Classes = np.unique(y).size
8     Input_shape = (9, 1)
9
10    # Change labels to a one-hot encoding format.
11    y = keras.utils.to_categorical(y - 1, num_classes=Classes)
12
13    # Split data to train test with the corresponding label.
14    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
15
16    return x_train, x_test, y_train, y_test, Classes, Input_shape
17 def load_cnn(classes, input_shape):
18     # Create the model based on the VGG-16
19     model = Sequential()
20     model.add(Conv1D(input_shape=input_shape, filters=64, kernel_size=3,
21         padding="same", activation="relu"))
22     model.add(Conv1D(filters=32, kernel_size=3, padding="same", activation="
23         relu"))
24     model.add(MaxPool1D(pool_size=2, strides=2))
25     model.add(Conv1D(filters=64, kernel_size=3, padding="same", activation="
26         relu"))
27     model.add(Conv1D(filters=64, kernel_size=3, padding="same", activation="
28         relu"))
29     model.add(MaxPool1D(pool_size=2, strides=2))
30     model.add(Flatten())
31     model.add(Dense(units=512, activation="relu"))
32     model.add(Dense(units=256, activation="relu"))
33     model.add(Dense(units=classes, activation="softmax"))
34
35     model.load_weights('CNN.h5')
36
37     return model
```

Source code 6: Load Model

```

1 def load_cnn_lstm(classes, input_shape):
2     # Define CNN model.
3     cnn = Sequential()
4     cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu', padding='same', input_shape=input_shape))
5     cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu', padding='same', input_shape=input_shape))
6     cnn.add(Dropout(0.5))
7     cnn.add(MaxPooling1D(pool_size=1))
8     cnn.add(Flatten())
9     # Define LSTM model.
10    model = Sequential()
11    model.add(TimeDistributed(cnn))
12    model.add(LSTM(units=100))
13    model.add(Dropout(0.5))
14    model.add(Dense(units=20, activation='relu'))
15    model.add(Dense(classes, activation='softmax'))
16
17    model.build(input_shape=(None, 9, 1))
18    model.load_weights('CNN_LSTM.h5')
19
20    return model
21
22 def load_cnn_gru(classes, input_shape):
23    model = Sequential()
24    model.add(Conv1D(64, 3, activation='relu', input_shape=input_shape))
25    model.add(SpatialDropout1D(0.2))
26    model.add(MaxPooling1D(2))
27    model.add(GRU(64))
28    model.add(Dropout(0.1))
29    model.add(Dense(classes, activation='softmax'))
30
31    model.load_weights('CNN_GRU.h5')
32
33    return model

```

Source code 7: Load Model

```

1 if __name__ == '__main__':
2     # Read the data.
3     X_Train, X_Test, Y_Train, Y_Test, classes, input_shape = load_data()
4
5     model_number = 2
6
7     if model_number == 1:
8         Model = load_cnn(classes, input_shape)
9     elif model_number == 2:
10        Model = load_cnn_lstm(classes, input_shape)
11
12    else:
13        Model = load_cnn_gru(classes, input_shape)
14    Model.summary()
15
16    # Change the optimization.
17    opt = Adam(learning_rate=0.001)
18    Model.compile(optimizer=opt, loss=keras.losses.categorical_crossentropy,
19                  metrics=['accuracy'])
20
21    predictions = Model.predict(X_Test)
22
23    score = Model.evaluate(X_Test, Y_Test, verbose=1)
24    print('Test loss:', score[0])
25    print('Test accuracy:', score[1])

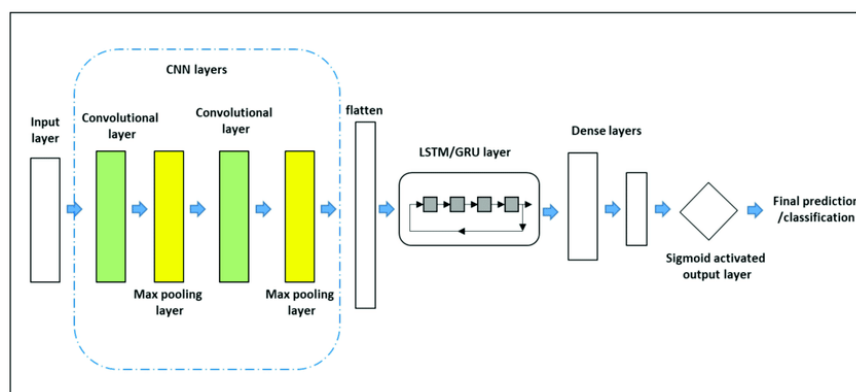
```

Source code 8: Load Model

5 Αισθητήρας στη θέση του χεριού – Πείραμα 1

Σκοπός του πειράματος είναι να βρεθεί αν θα χρησιμοποιηθεί ένας ή τρεις αισθητήρες και στη συνέχεια να διαπιστωθεί ποιο από τα τρία μοντέλα που χρησιμοποιούνται είναι το καλύτερο.

5.1 Αρχιτεκτονική των μοντέλων



Σχήμα 13: Αρχιτεκτονική και των 3 μοντελών[3]

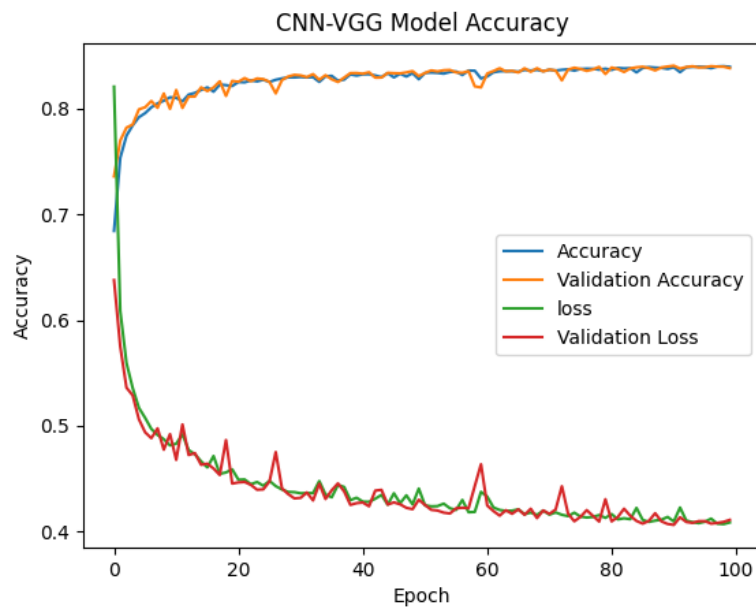
5.2 Μοντέλα με έναν αισθητήρα

5.2.1 Μοντέλο CNN

Αρχικά, πραγματοποιείται η κατάλληλη μορφοποίηση των δεδομένων. Έχοντας μόνο έναν αισθητήρα, το επιταχυνσόμετρο, και τα αρχεία έτοιμα, γίνεται μετατροπή του αρχείου με τα labels σε μορφή one-hot encoding. Του παρέχεται δηλαδή ένας πίνακας και το μετατρέπει σε μορφή άσσου και μηδέν. Τα επιθυμητά στοιχεία χαρακτηρίζονται με άσσο και τα μη επιθυμητά χαρακτηρίζονται με μηδέν. Έπειτα, γίνεται διαχωρισμός (split) στα δεδομένα σε εκπαίδευσης και δοκιμής.

Στη συνέχεια, ορίζεται το μοντέλο, το οποίο είναι ένα από τα πιο συνήθη και απλά, το sequential. Λειτουργεί με απλό τρόπο και για αυτό αποτελεί ένα χρήσιμο εργαλείο για την γρήγορη ανάπτυξη μοντέλων. Χρησιμοποιεί ένα ένα τα δεδομένα, τα οποία φιλτράρονται από κάθε στρώση. Το μοντέλο έχει 2 στρώσεις Conv1D, η μία με 64 και η άλλη με 32 φίλτρα. Επιπλέον, περιλαμβάνει kernel πυρήνα μεγέθους 3, padding “same”, ώστε να έχει η έξοδος ίδια διάσταση με την είσοδο και λειτουργία ενεργοποίησης ReLU. Το μοντέλο έχει μέγιστη υποδειγματοληψία (MaxPooling1D) με μέγεθος 2 και βήμα 2. Έχει άλλες 2 στρώσεις με Conv1D, ακριβώς το ίδιο με εκείνη με τα 64 φίλτρα. Μία ακόμα υποδειγματοληψία και έτσι προστίθεται ένα επίπεδο συνέλιξης Flatten. Τέλος, πριν την έξοδο, υπάρχουν 2 πλήρως συνδεδεμένα επίπεδα (Dense) με 512 νευρώνες και 256 νευρώνες και ενεργοποίηση ReLU αντίστοιχα. Η έξοδος γίνεται με ένα τελευταίο επίπεδο, το οποίο περιέχει τον αριθμό των κλάσεων για μονάδες εξόδου, ενώ για συνάρτηση ενεργοποίησης έχει την SoftMax, η οποία χρησιμεύει στην ταξινόμηση.

Μετά τη δημιουργία του μοντέλου με τα φίλτρα του, πραγματοποιείται βελτιστοποίηση (optimize) με σκοπό την επιβεβαίωση ύπαρξης σταθερότητας στις παραμέτρους κατά την διάρκεια της εκπαίδευσης. Τέλος, το μοντέλο σχεδιάζεται με τέτοιο τρόπο ώστε να είναι σε θέση να ολοκληρώνει γρηγορότερα – σε περίπτωση που βρει πολύ καλό αποτέλεσμα – και η εκπαίδευσή του γίνεται για 100 εποχές και για κάθε πόσα δεδομένα του εκχωρούνται, δηλαδή 10000. Γίνεται εξαγωγή των αποτελεσμάτων σε ένα plot και διαπιστώνεται η πρόοδος του μοντέλου με όλα τα φίλτρα που του έχουν εισαχθεί.

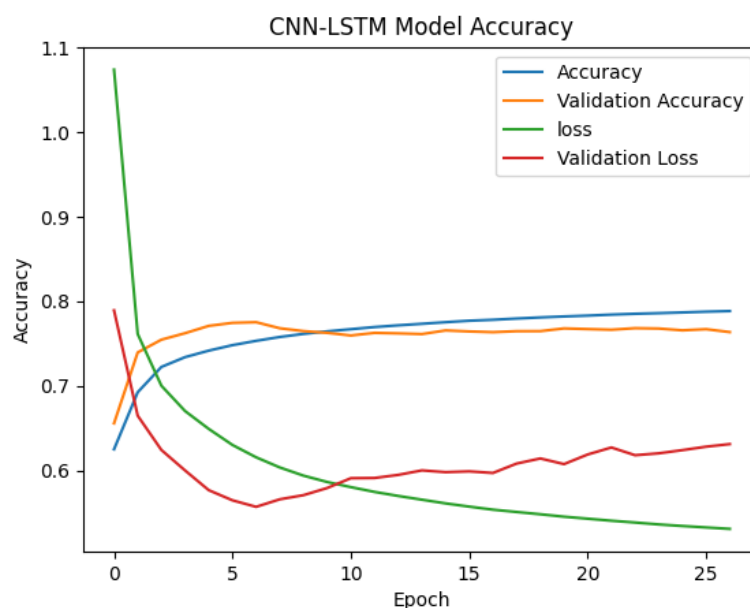


Σχήμα 14: plot CNN-VGG

5.2.2 Μοντέλο CNN-LSTM

Σε αυτό το μοντέλο, ο κώδικας που έχει φτιαχτεί ουσιαστικά ορίζει ένα σύνθετο νευρωνικό δίκτυο που συνδυάζει το CNN και LSTM (Long Short-Term Memory) layers για να διαπιστωθεί αν είναι καλύτερο από το απλό CNN μοντέλο. Παρόμοια με τον κώδικα στο CNN, ο κώδικας είναι παρόμοιος με την μόνη διαφορά να έγκειται στον τρόπο δημιουργίας των μοντέλων. Για το πρώτο μοντέλο, το CNN, προστίθενται 2 στρώσεις Conv1D, όπου κάθε στρώση έχει 64 φίλτρα με kernel πυρήνα μεγέθους 3 και την λειτουργία ενεργοποίησης ReLU. Έπειτα, ορίζεται μια στρώση Dropout, που απενεργοποιεί τυχαία μια προκαθορισμένη ποσότητα νευρώνων για αποτροπή υπερεκπαίδευσης. Τέλος, για το πρώτο μοντέλο υπάρχει μια στρώση MaxPooling με μείωση διαστάσεων μεγέθους 1 και η συνάρτηση Flatten.

Ορίζεται το νευρωνικό δίκτυο LSTM, όπου δημιουργείται μια στρώση TimeDistributed που εφαρμόζει το πρώτο μοντέλο σε κάθε βήμα της εισόδου. Στην συνέχεια, προστίθεται μια στρώση LSTM με 100 μονάδες, και μία στρώση Dropout που σκοπεύει στη μείωση του υπερεκπαιδευτικού φαινομένου. Πριν την έξοδο, υπάρχουν 2 στρώσεις Dense, όπου η μια έχει 20 μονάδες και συνάρτηση ενεργοποίησης ReLU και η δεύτερη στρώση παίρνει τον αριθμό των κλάσεων με συνάρτηση ενεργοποίησης την SoftMax. Όπως ακριβώς και με το CNN μοντέλο, πραγματοποιείται και σε αυτό εκπαίδευση με τα ίδια ακριβώς στοιχεία και εξάγεται το plot με τα αποτελέσματα.

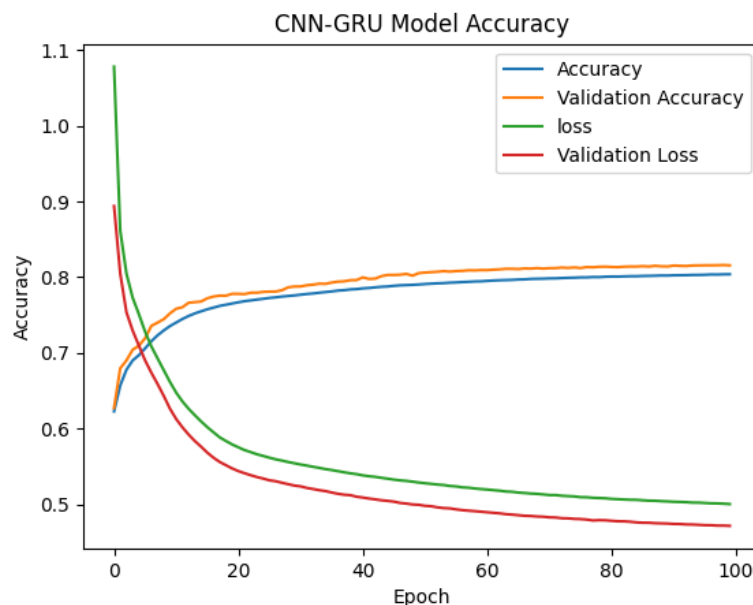


Σχήμα 15: plot CNN-LSTM

5.2.3 Μοντέλο CNN-GRU

Ομοίως και σε αυτό το μοντέλο, αλλάζει κατά κύριο τμήμα το πώς ορίζεται το μοντέλο. Εδώ υπάρχει το μοντέλο CNN και το μοντέλο GRU (Gated Recurrent Units). Το GRU είναι, ουσιαστικά, τύπος αναδρομικών νευρωνικών μονάδων που αντιμετωπίζουν τα δεδομένα αναδρομικά και για να διατηρούν χρονικά συνεχόμενα τις πληροφορίες.

Ξεκινώντας, το μοντέλο έχει μια στρώση Conv1D με 64 φίλτρα, με απλό πυρήνα μεγέθους 3, συνάρτηση ενεργοποίησης ReLU και δέχεται σαν είσοδο 3 χρονικά βήματα για κάθε 1 χαρακτηριστικό. Η δεύτερη στρώση είναι μια SpatialDropout1D με 0.2 ποσοστό. Εδώ, ουσιαστικά, απενεργοποιούνται τυχαία οι νευρωνικές μονάδες στα χρονικά βήματα, προκειμένου να αποτραπεί η υπερεκπαίδευση. Έπειτα, έχει μια στρώση MaxPooling1D με μέγεθος 1. Μετά από αυτήν την στρώση, ορίζεται το GRU με 64 νευρώνες. Έχει ένα Dropout με ποσοστό 0.1 και μία τελική στρώση Dense, και – όπως και στα προηγούμενα – μονάδες εξόδου τον αριθμό των κλάσεων και συνάρτηση ενεργοποίησης την SoftMax. Παρακάτω, πραγματοποιείται εκπαίδευση στο μοντέλο και εξάγονται τα αποτελέσματα για την συγκεκριμένη συνέλιξη.



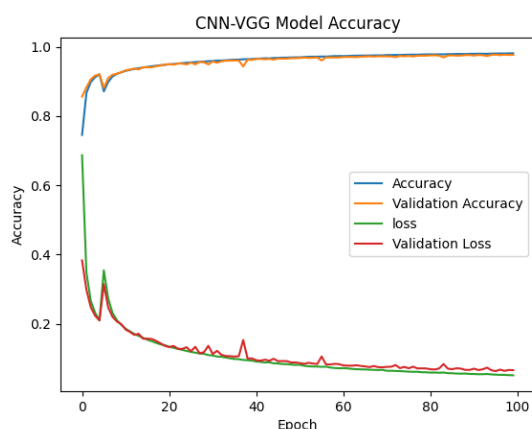
Σχήμα 16: plot CNN-GRU

5.3 Μοντέλα με τρεις αισθητήρες

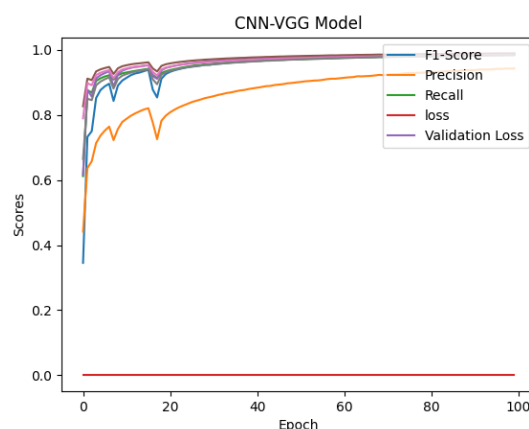
Και στα τρία μοντέλα προστέθηκαν 2 ακόμη αισθητήρες, το μαγνητόμετρο και το γυροσκόπιο. Το κάθε μοντέλο παραμένει το ίδιο ακριβώς στην διαδικασία. Πραγματοποιήθηκαν μερικές αλλαγές, αρχικά στο preprocessing, όπου στη συνάρτηση με την κανονικοποίηση το range αλλάζει σε 9, εξαιτίας του γεγονότος ότι κάθε αισθητήρας έχει 3 άξονες (x, y, z). Έπειτα, στο data-handler διαφέρουν οι στήλες οι οποίες γίνονται drop στο κάθε αρχείο. Τέλος, στο κάθε μοντέλο αλλάζει το input-shape, δηλαδή δίνεται 9 για την είσοδο – σε αντίθεση με 3 που δινόταν στον έναν αισθητήρα - για τον ίδιο λόγο που αλλάζει και το preprocessing και το data-handler.

5.3.1 Μοντέλο CNN

Η διαδικασία είναι ακριβώς ίδια όπως και με τον έναν αισθητήρα, με μοναδική διαφορά πως επιτεύχθηκαν οι απαραίτητες αλλαγές για την προσθήκη των 2 ακόμη αισθητήρων. Τα αποτελέσματα βγαίνουν ομοίως σε plot διάγραμμα.



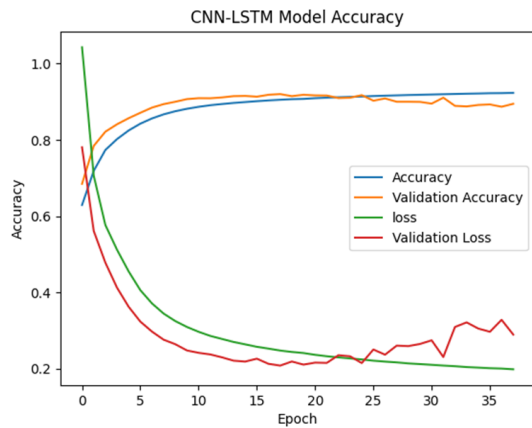
Σχήμα 17: plot CNN-VGG accuracy



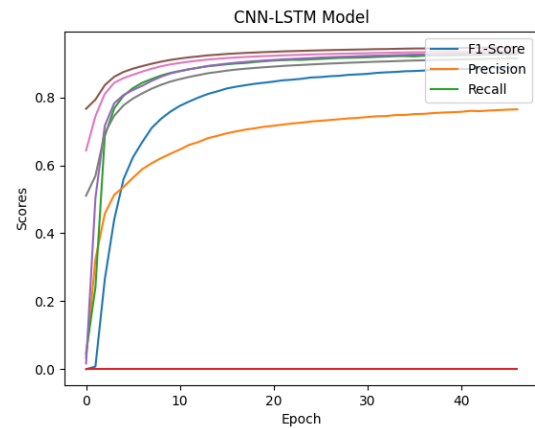
Σχήμα 18: Plot CNN-VGG scores

5.3.2 Μοντέλο CNN-LSTM

Οι ίδιες αλλαγές ισχύουν και για αυτό το μοντέλο. Όλα τα φίλτρα και τα συναφή παραμένουν ίδια, και αλλάζουν τα καινούρια που αναφέρθηκαν για τους 2 επιπλέον αισθητήρες. Τα αποτελέσματα βγαίνουν σε ένα plot διάγραμμα.



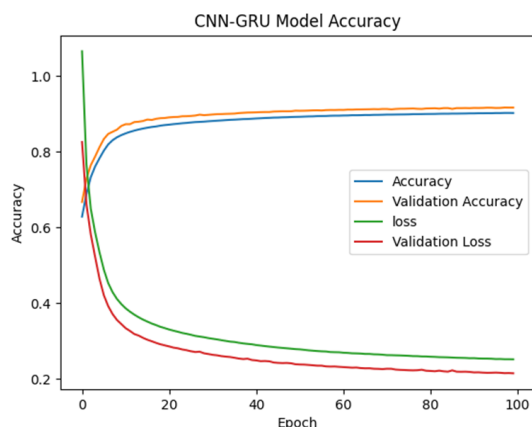
Σχήμα 19: plot CNN-LSTM accuracy



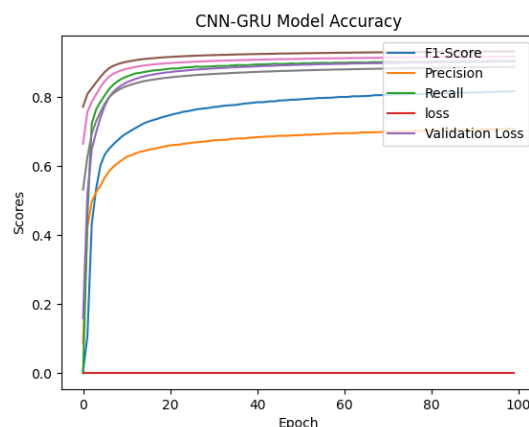
Σχήμα 20: Plot CNN-LSTM scores

5.3.3 Μοντέλο CNN-GRU

Παρόμοια και στο τρίτο μοντέλο, πραγματοποιήθηκαν ίδιες αλλαγές στα συγκεκριμένα σημεία. Όλα τα υπόλοιπα στοιχεία παραμένουν ως έχουν. Το αποτέλεσμα βγαίνει σε ένα plot διάγραμμα.



Σχήμα 21: Plot CNN-GRU accuracy



Σχήμα 22: Plot CNN-GRU scores

5.4 Αποτελέσματα

Η απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης μπορεί να φανεί μέσω ενός γραφήματος, στο οποίο αποτυπώνεται ο άξονας του χρόνου ή των επαναλήψεων (epochs) στον άξονα x, και η απώλεια και η ευστοχία στον άξονα y.

Η Απώλεια (Loss) αφορά τη μετρική που χρησιμοποιείται για να εκφραστεί η διαφορά των προβλέψεων του μοντέλου από τις πραγματικές τιμές. Η μείωση της απώλειας επιτυγχάνεται μέσω της εκπαίδευσης του μοντέλου. Η Απώλεια επικύρωσης (Validation loss) αναφέρεται στην απώλεια υπολογισμένη στα δεδομένα επικύρωσης και η χρήση του επικεντρώνεται στην αξιολόγηση της δυνατότητας γενίκευσης του μοντέλου σε νέα δεδομένα. Αύξηση του Validation loss με ταυτόχρονη μείωση του Loss μπορεί να σχετίζεται με υπερεκπαίδευση.

Η ευστοχία (Accuracy) αναφέρεται στο ποσοστό των σωστών προβλέψεων που κάνει το μοντέλο σε σύνολο δεδομένων. Η ευστοχία επικύρωσης (Validation accuracy) αφορά το ποσοστό των σωστών προβλέψεων στα δεδομένα επικύρωσης και χρησιμοποιείται για τη μέτρηση της δυνατότητας γενίκευσης του μοντέλου. Μείωση της Validation accuracy μπορεί να προκληθεί όταν το μοντέλο ξεκινά τη γενίκευση κατά την εκπαίδευση.

Γενικά, το γράφημα της απώλειας και της επικύρωσης επιτυγχάνουν την παρακολούθηση της απόδοσης του μοντέλου κατά τη διάρκεια της εκπαίδευσης, με σκοπό τη λήψη αποφάσεων σχετικά με το σημείο λήξης της εκπαίδευσης ή το χρονικό όριο προσαρμογής των παραμέτρων του μοντέλου.

5.4.1 Αποτελέσματα από έναν αισθητήρα

Loss – Validation loss

Στο γράφημα του CNN παρατηρείται αρχικά υψηλή τιμή του Loss, εξαιτίας της απουσίας εκπαίδευσης του μοντέλου και της έλλειψης δεδομένων. Κατά τη διάρκεια της διαδικασίας εκπαίδευσης, προστίθενται δεδομένα και αρχίζει να μειώνεται και το Loss. Σε περίπτωση υπερεκπαίδευσης του μοντέλου, παρατηρούνται spikes – εμφανίζονται εντονότερα στην καμπύλη του Validation loss – τα οποία μειώνονται όταν εισάγονται νέα δεδομένα, λόγω της δυνατότητας γενίκευσης του μοντέλου. Αντίστοιχα στο γράφημα του CNN-LSTM, η καμπύλη του Loss ξεκινά από υψηλή τιμή και μειώνεται ομαλά. Η διαφορά εντοπίζεται στην καμπύλη του Validation loss, όπου παρατηρείται αύξηση λόγω της διαρκούς υπερεκπαίδευσης του μοντέλου κατά την προσθήκη νέων παρόμοιων δεδομένων. Με παρόμοιο τρόπο εμφανίζονται και οι καμπύλες στο γράφημα CNN-GRU, οι οποίες ωστόσο μειώνονται με πολύ ομαλό τρόπο.

Accuracy – Validation accuracy

Η καμπύλη της ευστοχίας, στο γράφημα CNN, ξεκινά από μια μέση τιμή και ακολουθεί μια ομαλή ανοδική πορεία, η οποία εμφανίζει ήπια spikes, εξαιτίας της γενίκευσης των δεδομένων. Στο γράφημα CNN-LSTM, η καμπύλη της Accuracy ακολουθεί παρόμοια πορεία. Η μόνη διαφορά έγκειται στο γεγονός ότι υπάρχει υπερεκπαίδευση του μοντέλου, οπότε η καμπύλη της Validation accuracy έχει μια ελαφρώς καθοδική πορεία. Ομοίως, οι καμπύλες στο γράφημα CNN-GRU αποτυπώνονται με ομαλή ανοδική πορεία.

5.4.2 Αποτελέσματα από τρεις αισθητήρες

Loss – Validation loss

Τα αποτελέσματα από τους τρεις αισθητήρες εμφανίζονται σε αναλογία με τα αποτελέσματα από τον έναν αισθητήρα, με κάποιες μικρές διαφορές, οι οποίες αφορούν το γεγονός ότι το Loss είναι μικρότερο όταν πρόκειται για τους τρεις αισθητήρες.

Accuracy – Validation accuracy

Αντίστοιχα, και σε αυτή την περίπτωση οι καμπύλες των γραφημάτων των μοντέλων με τους τρεις αισθητήρες ακολουθούν παρόμοια μορφή με εκείνες των μοντέλων ενός αισθητήρα. Η σημαντικότερη, ωστόσο, διαφορά που προκύπτει είναι πως η Accuracy στα μοντέλα τριών αισθητήρων είναι εμφανώς μεγαλύτερη.

5.4.3 Επιπλέον μετρικές

Στο γράφημα του CNN, το Precision και το Recall ξεκινούν από μια μέση τιμή, ενώ το F1 score από χαμηλότερη τιμή, και κατά τη διάρκεια της εκπαίδευσης οι καμπύλες του Recall και του F1 score φτάνουν σχεδόν τη μέγιστη τιμή, ενώ η καμπύλη του Precision φτάνει σε μια αρκετά ικανοποιητική τιμή.

Το γράφημα του CNN – LSTM εμφανίζει αρκετές ομοιότητες με το CNN – GRU, με μοναδική διαφορά μικρές μεταβολές στις καμπύλες των μετρικών τους, όπου οι τελικές τιμές του CNN – LSTM είναι συγκριτικά υψηλότερες. Επιπλέον, το μοντέλο του CNN – LSTM τερματίζεται νωρίτερα χρονικά, εξαιτίας της υπερεκπαίδευσης.

5.4.4 Σύγκριση αποτελεσμάτων

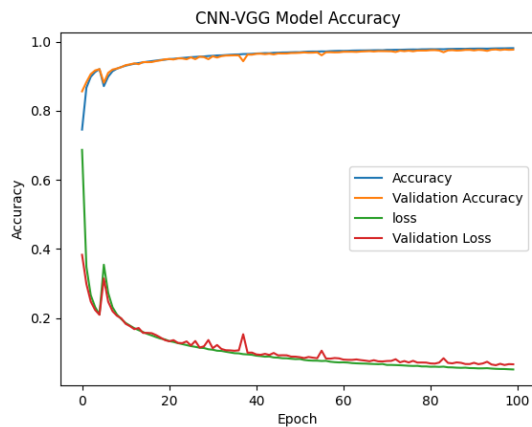
Με βάση τα αποτελέσματα του πειράματος 1, προκύπτει το συμπέρασμα πως είναι προτιμότερη η χρήση μοντέλων με τρεις αισθητήρες διότι τα αποτελέσματα που προκύπτουν είναι πιο σταθερά και ακριβή. Επίσης, με βάση τις επιπλέον μετρικές που εφαρμόστηκαν, διαπιστώθηκε ότι το ιδανικότερο μοντέλο για τη συνέχιση της εργασίας είναι το CNN.

6 Αισθητήρες στα υπόλοιπα σημεία – Πείραμα 2

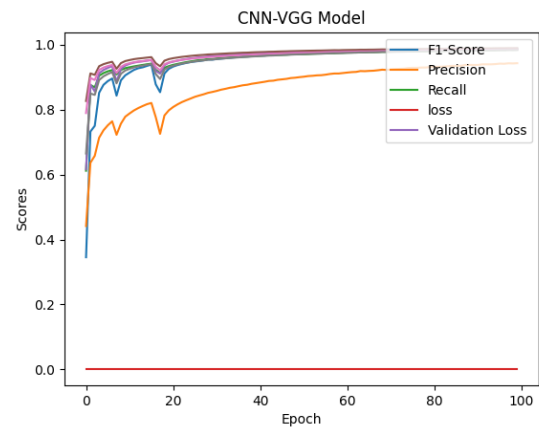
6.1 Πρώτος χρήστης

6.1.1 Αισθητήρες στο χέρι

Με βάση τα αποτελέσματα του πειράματος 1, αποδείχθηκε ότι για τους αισθητήρες που τοποθετήθηκαν στο χέρι, το καλύτερο μοντέλο είναι το CNN.



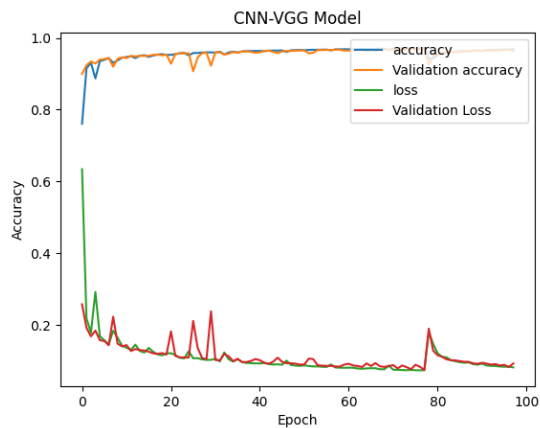
Σχήμα 23: Plot hand accuracy user1



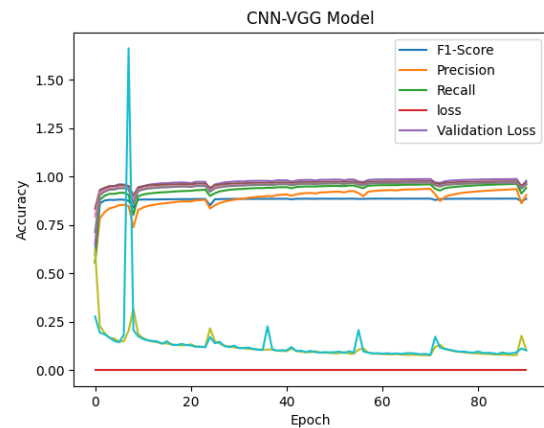
Σχήμα 24: Plot hand scores user1

6.1.2 Αισθητήρες στην τσάντα

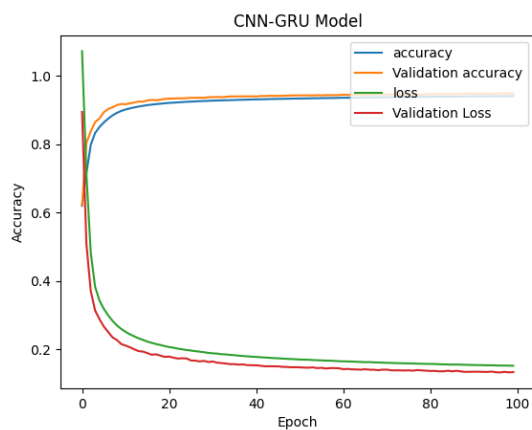
Για να διαπιστωθεί αν προκύπτουν διαφορετικά αποτελέσματα σε σχέση με εκείνα του πρώτου πειράματος, δοκιμάστηκαν δύο από τα τρία μοντέλα (CNN και CNN – GRU), όπου αποδείχθηκε ότι το μοντέλο CNN έχει σχεδόν άριστο αποτέλεσμα, σε αντίθεση με το CNN – GRU που παραμένει αρκετά καλό αλλά όχι τόσο άρτιο.



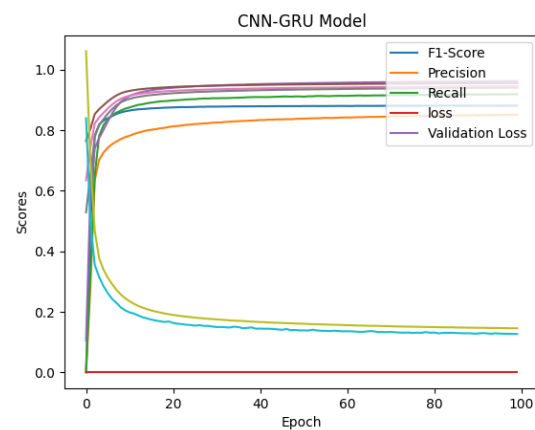
Σχήμα 25: Plot CNN bag accuracy user1



Σχήμα 26: Plot CNN bag scores user1



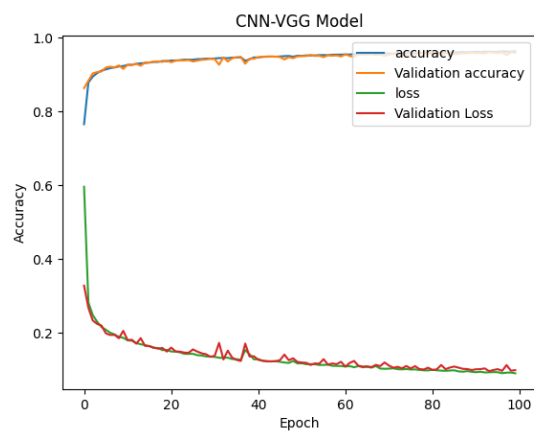
Σχήμα 27: Plot GRU bag accuracy user1



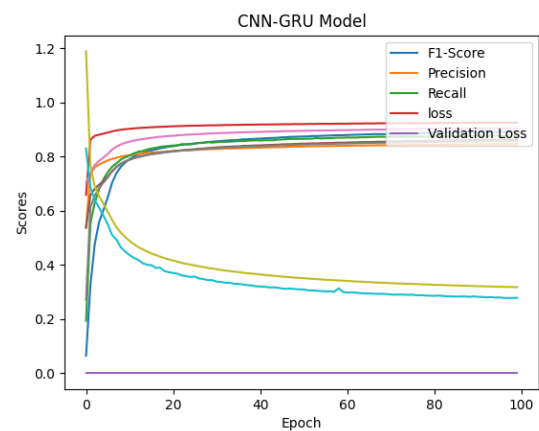
Σχήμα 28: Plot GRU bag scores user1

6.1.3 Αισθητήρες στο στήθος

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



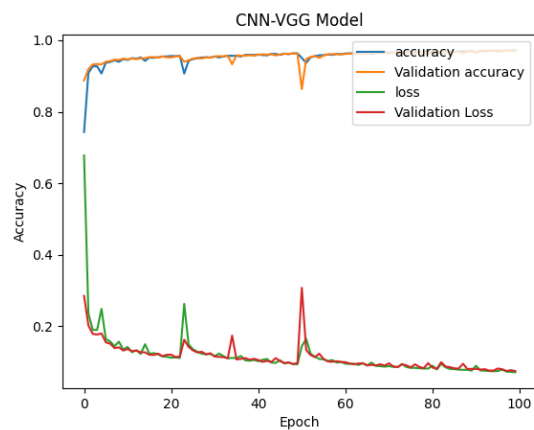
Σχήμα 29: Plot torso accuracy user1



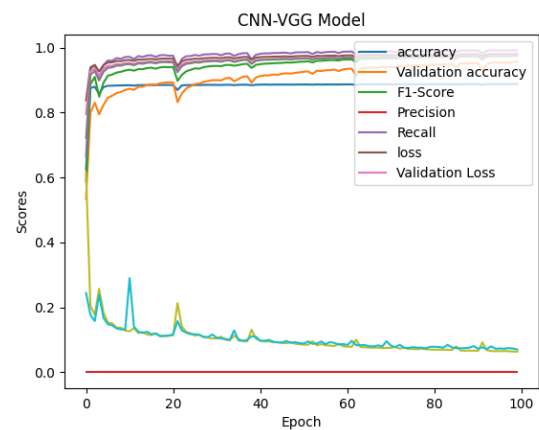
Σχήμα 30: Plot torso scores user1

6.1.4 Αισθητήρες στην πρόσθια τσέπη

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



Σχήμα 31: Plot hips accuracy user1

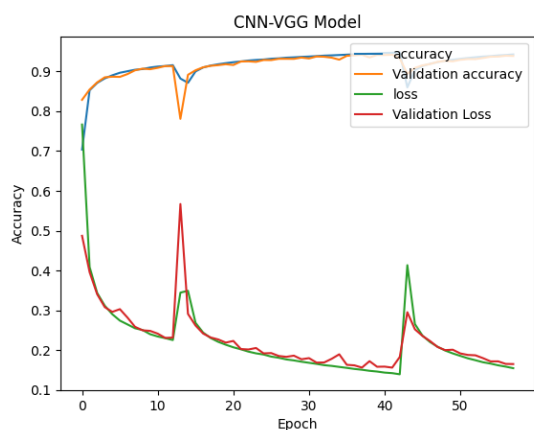


Σχήμα 32: Plot hips scores user1

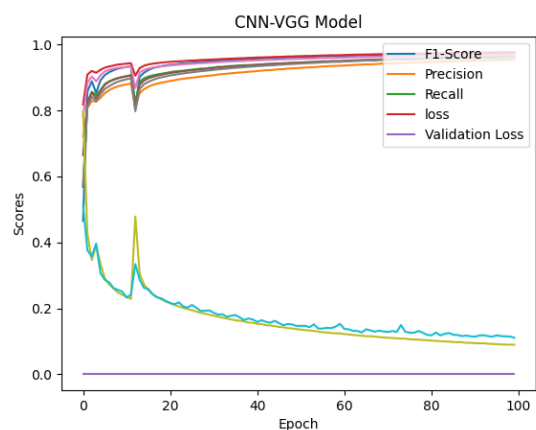
6.2 Δεύτερος χρήστης

6.2.1 Αισθητήρες στο χέρι

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



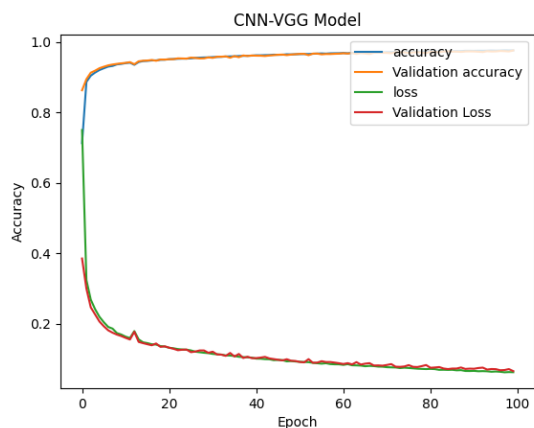
Σχήμα 33: Plot hand accuracy user2



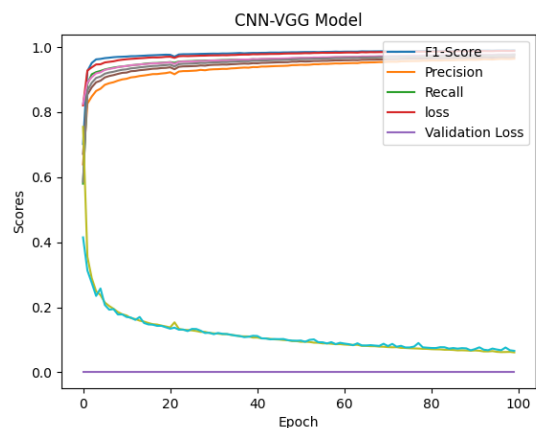
Σχήμα 34: Plot hand scores user2

6.2.2 Αισθητήρες στην τσάντα

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



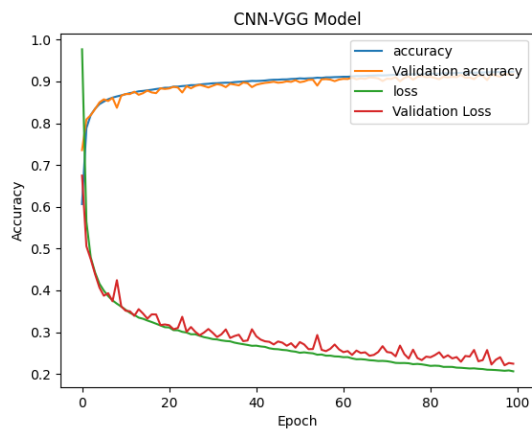
Σχήμα 35: Plot bag accuracy user2



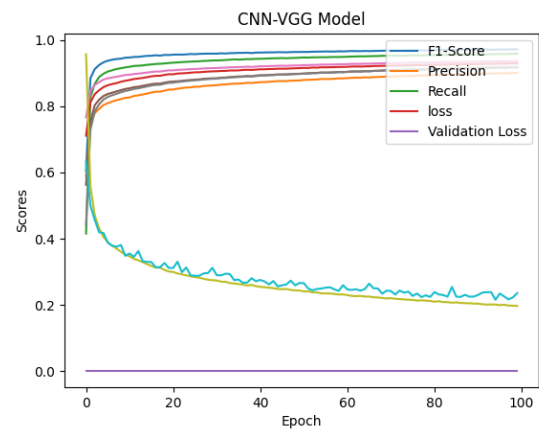
Σχήμα 36: Plot bag scores user2

6.2.3 Αισθητήρες στο στήθος

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



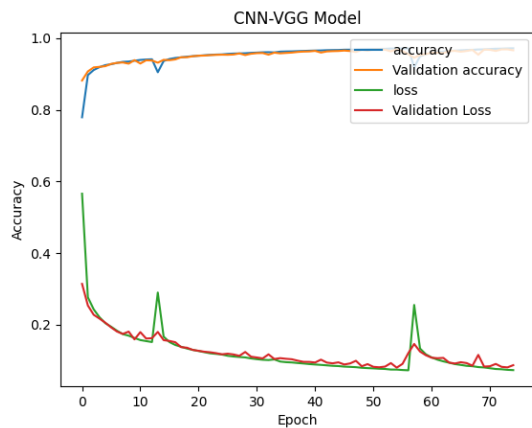
Σχήμα 37: Plot torso accuracy user2



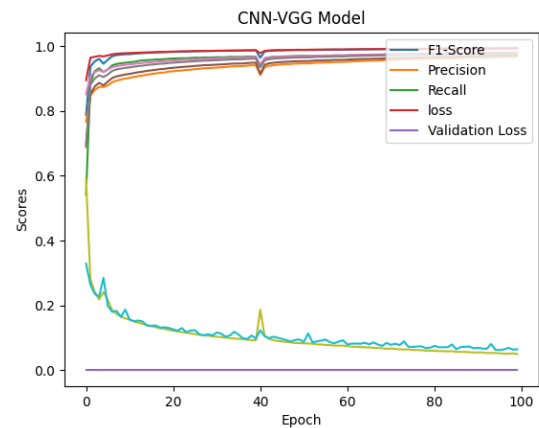
Σχήμα 38: Plot torso scores user2

6.2.4 Αισθητήρες στην πρόσθια τσέπη

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



Σχήμα 39: Plot hips accuracy user2

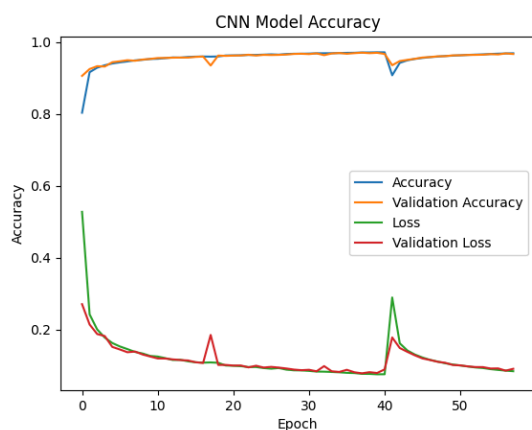


Σχήμα 40: Plot hips scores user2

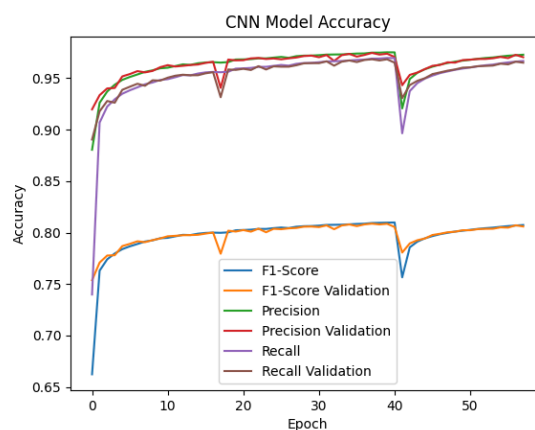
6.3 Τρίτος χρήστης

6.3.1 Αισθητήρες στο χέρι

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



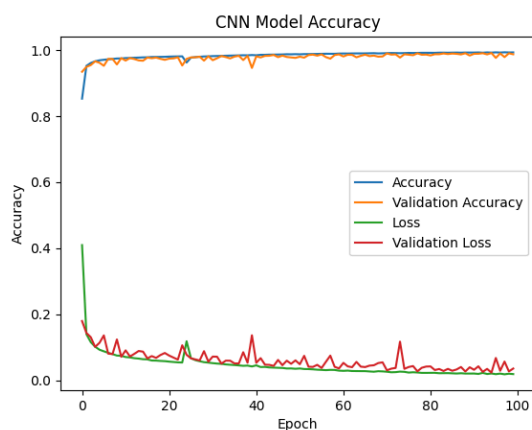
Σχήμα 41: Plot hand accuracy user3



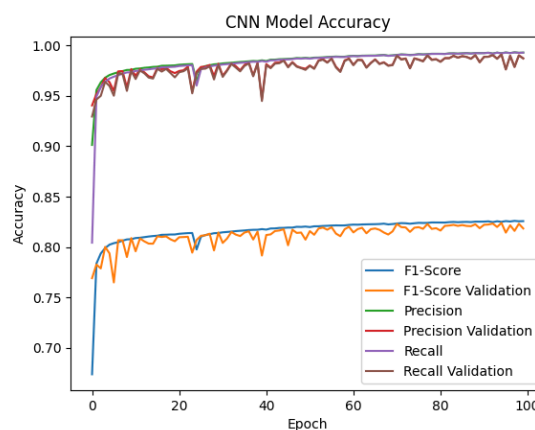
Σχήμα 42: Plot hand scores user3

6.3.2 Αισθητήρες στην τσάντα

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



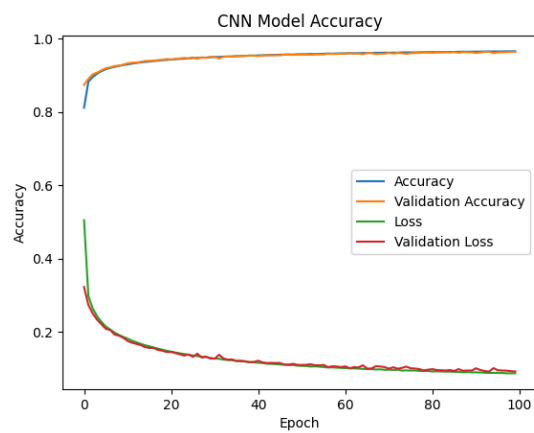
Σχήμα 43: Plot bag accuracy user3



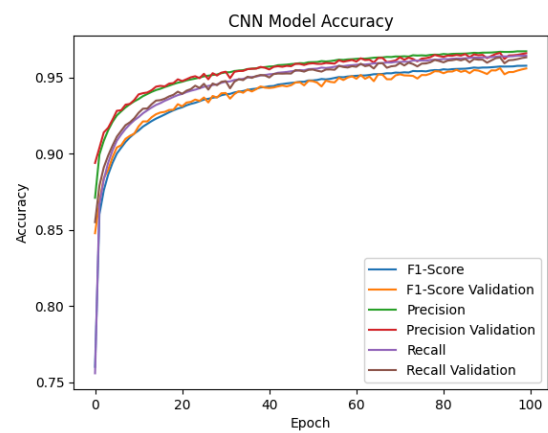
Σχήμα 44: Plot bag scores user3

6.3.3 Αισθητήρες στο στήθος

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



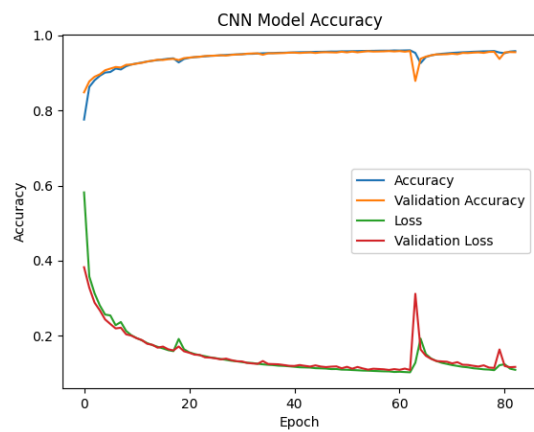
Σχήμα 45: Plot torso accuracy user3



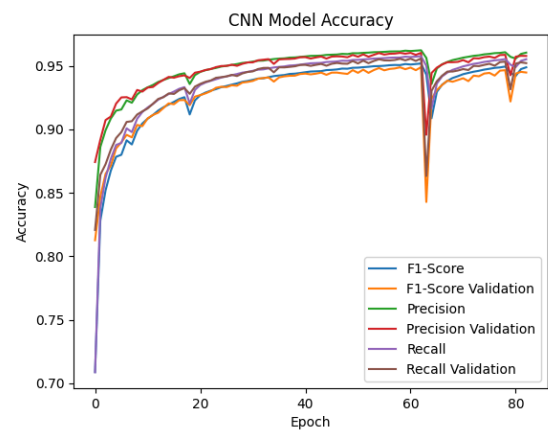
Σχήμα 46: Plot torso scores user3

6.3.4 Αισθητήρες στην πρόσθια τσέπη

Το μοντέλο που χρησιμοποιήθηκε είναι το CNN.



Σχήμα 47: Plot hips accuracy user3



Σχήμα 48: Plot hips scores user3

6.4 Σύγκριση αποτελεσμάτων

6.4.1 Πρώτος χρήστης

Διαπιστώθηκε, με βάση τα διαγράμματα, πως το accuracy και το F1 score είχε αρκετά καλά αποτελέσματα σε όλες τις θέσεις. Ωστόσο, παρατηρείται ότι στη θέση του χεριού δεν υπήρχαν σχεδόν καθόλου μεταβολές σε όλο το χρονικό περιθώριο, σε αντίθεση με τις θέσεις της τσάντας και του στήθους, όπου υπάρχουν κάποιες μικρές μεταβολές. Οι περισσότερες αναταράξεις εμφανίζονται στη θέση της τσέπης, διότι δεν υπήρχε αρκετή σταθερότητα κατά τη διάρκεια των δραστηριοτήτων.

6.4.2 Δεύτερος χρήστης

Στην περίπτωση αυτή, παρατηρείται πως στις θέσεις της τσάντας και της τσέπης εμφανίζονται πολύ ομαλά αποτελέσματα, στη θέση του στήθους λιγότερο ομαλά, ενώ στη θέση του χεριού υπάρχει μεγάλη μετατόπιση κατά τη διάρκεια μιας χρονικής στιγμής και στη συνέχεια παρουσιάζεται πιο ομαλή κίνηση.

6.4.3 Τρίτος χρήστης

Τα αποτελέσματα είναι ικανοποιητικά και σε αυτή την περίπτωση, με τα πιο ομαλά αποτελέσματα να εμφανίζονται στη θέση του στήθους, τα αμέσως λιγότερο ομαλά στις θέσεις του χεριού και της τσέπης, ενώ στη θέση της τσάντας παρουσιάζονται μικρές μεταβολές καθ' όλη τη διάρκεια των δραστηριοτήτων.

7 Leave-one-subject-out

7.1 Έννοια

Το leave-one-subject-out[10][16] είναι μία μέθοδος που χρησιμοποιείται στην μηχανική μάθηση για ανάλυση δεδομένων και την εκτίμηση αυτών. Αρχικά, διαχωρίζονται τα δεδομένα σε υποσύνολα όπου το καθένα αντιστοιχεί σε μία κατηγορία – στην προκειμένη περίπτωση οι χρήστες. Έπειτα, για κάθε διαδικασία εκτίμησης, όπως την εκπαίδευση μοντέλου και αξιολόγηση απόδοσης, επιλέγεται ένα υποκείμενο ως test set ενώ τα υπόλοιπα αφορούν την εκπαίδευση του μοντέλου. Αυτό γίνεται για όλα τα υποκείμενα για να αξιολογηθούν οι αποδόσεις των μοντέλων/ μοντέλου σε κάθε υποκείμενο.

Σκοπός είναι η ανταπόκριση του μοντέλου μηχανικής μάθησης με διαφορετικά υποκείμενα. Μπορεί να αξιολογηθεί, έτσι, η ικανότητα του μοντέλου στην γενίκευση νέων δεδομένων αφού κάθε υποκείμενο αντιπροσωπεύει μια διαφορετική πηγή ποικιλίας. Βέβαια, θα πρέπει να ληφθεί υπόψη πως μπορεί να υπάρξει μεγάλη ανισορροπία δεδομένων και αύξηση του φόρτου. Τέλος, αν είναι ανάγκη να αξιολογηθούν οι αποδόσεις των μοντέλων, η συγκεκριμένη τεχνική είναι ένα πολύ ισχυρό εργαλείο.

7.2 Παρόν πείραμα

Γίνεται run του μοντέλου CNN για όλους τους χρήστες, βγάζοντας εκτός τον χρήστη 1, άρα τα δεδομένα του θα αφορούν το test data και γίνεται εκπαίδευση για τους χρήστες 2 και 3. Επίσης, τα δεδομένα που συγχωνεύτηκαν, αποθηκεύτηκαν σε ένα αρχείο .h5 για να γίνει έπειτα η διαδικασία LOOCV. Αντίστοιχα, γίνεται η ίδια διαδικασία και για τους υπόλοιπους δύο χρήστες διαδοχικά. Λόγω του χρόνου που χρειάζεται για κάθε εκπαίδευση, το train γίνεται για έως 20 εποχές για κάθε χρήστη.

```

1 # Function for normalizing data in each column but not the label column.
2 def normalize(df):
3     for k in range(9):
4         df[k] = preprocessing.MinMaxScaler(feature_range=(-1, 1)).fit_transform(
5             np.array(df[k]).reshape(-1, 1))
6     return df
7
8 if __name__ == '__main__':
9     # To read data for all users or only 1.
10    use_multiple_users = 1
11    if use_multiple_users == 1:
12        range_size = 4
13    else:
14        range_size = 1
15
16    # Loop for Users.
17    for i in range(1, range_size):
18        # Loop for Days.
19        for j in range(1, 4):
20            file_path = 'Data/' + 'User' + str(i) + '/Day' + str(j) # Set
21            filepath
22
23            phone_position = 'Hand' # Set phone position
24            if phone_position == 'Hand':
25                # Load motion data and labels.
26                print('Loading Hand motion data for User ' + str(i) + ' Day ' + str(
27                    j) + '\n')
28                data_df = pd.read_csv(file_path + '/Hand_Motion.txt',
29                    delim_whitespace=True, header=None)
30            elif phone_position == 'Torso':
31                # Load motion data and labels.
32                print('Loading Torso motion data for User ' + str(i) + ' Day ' + str(
33                    j) + '\n')
34                data_df = pd.read_csv(file_path + '/Torso_Motion.txt',
35                    delim_whitespace=True, header=None)
36            elif phone_position == 'Bag':
37                # Load motion data and labels.
38                print('Loading Bag motion data for User ' + str(i) + ' Day ' + str(
39                    j) + '\n')
40                data_df = pd.read_csv(file_path + '/Bag_Motion.txt',
41                    delim_whitespace=True, header=None)
42            else:
43                # Load motion data and labels.
44                print('Loading Hips motion data for User ' + str(i) + ' Day ' + str(
45                    j) + '\n')
46                data_df = pd.read_csv(file_path + '/Hips_Motion.txt',
47                    delim_whitespace=True, header=None)
48
49            print('Loading label data for User ' + str(i) + ' Day ' + str(j) + '\n
50            ')

```

```

40     label_df = pd.read_csv(file_path + '/Label.txt', delim_whitespace=True
    , header=None)
41
42     # Drop unnecessary columns.
43     label_df = label_df.drop(label_df.iloc[:, 2:8], axis=1)
44     label_df = label_df.drop(label_df.columns[[0]], axis=1)
45     data_df = data_df.drop(data_df.iloc[:, 10:23], axis=1)
46     data_df = data_df.drop(data_df.columns[[0]], axis=1)
47     # Concatenate the dataframes.
48     data_df = pd.concat([data_df, label_df], ignore_index=True, sort=False
    , axis=1, join='inner')
49     # Drop NaN rows.
50     data_df = data_df.dropna()
51     # Normalize data between -1 and 1.
52     data_df = normalize(data_df)
53     # Write the final dataframe.
54     data_df.to_csv(file_path + '/Preprocessed_Data.csv', index=False, sep=
    ';', header=False)

```

Source code 9: Preprocessing code LOOCV

```

1  if __name__ == '__main__':
2      # To read data for all users or only 1.
3      use_multiple_users = 1
4      if use_multiple_users == 1:
5          range_size = 4
6      else:
7          range_size = 2
8
9      # Loop for Users.
10     for i in range(1, range_size):
11         # Loop for Days.
12         for j in range(1, 4):
13             file_path = 'Data/' + 'User' + str(i) + '/Day' + str(j)
14             # Load motion data and labels.
15             print('Loading data for User ' + str(i) + ' Day ' + str(j) + '\n')
16             df = pd.read_csv(file_path + '/Preprocessed_Data.csv', sep=';', header
    =None)
17             # df_x = df.drop(df.columns[[3, 4, 5, 6, 7, 8, 9]], axis=1) Uncomment
    to use 1 sensor.
18             # Use 3 sensors.
19             df_x = df.drop(df.columns[[9]], axis=1)
20             # Write the final dataframe.
21             df_x.to_csv(file_path + '/Data.csv', index=False, sep=';', header=
    False)
22             df_y = df.drop(df.columns[[0, 1, 2, 3, 4, 5, 6, 7, 8]], axis=1)
23             # Write the final dataframe.
24             df_y.to_csv(file_path + '/Labels.csv', index=False, sep=';', header=
    False)

```

Source code 10: Data Handler code LOOCV


```

1 def save_plots(history, acc):
2     if acc == 1:
3         fig, ax = plt.subplots()
4         # Plot and save the Model accuracy.
5         ax.plot(history.history["accuracy"])
6         ax.plot(history.history['val_accuracy'])
7         ax.plot(history.history['loss'])
8         ax.plot(history.history['val_loss'])
9         ax.set_title("CNN Model Accuracy")
10        ax.set_ylabel("Accuracy")
11        ax.set_xlabel("Epoch")
12        ax.legend(["Accuracy", "Validation Accuracy", "Loss", "Validation Loss"
13        ])
14        fig.savefig('Images/Model_Accuracy_CNN_VGG_3.png')
15        fig.clf()
16        ax.cla()
17        del fig, ax
18    else:
19        fig, ax = plt.subplots()
20        ax.plot(history.history["f1_score"])
21        ax.plot(history.history["val_f1_score"])
22        ax.plot(history.history['precision'])
23        ax.plot(history.history['val_precision'])
24        ax.plot(history.history['recall'])
25        ax.plot(history.history['val_recall'])
26        ax.set_title("CNN Model Accuracy")
27        ax.set_ylabel("Scores")
28        ax.set_xlabel("Epoch")
29        ax.legend(["F1-Score", "F1-Score Validation", "Precision", "Precision
30        Validation", "Recall", "Recall Validation"])
31        fig.savefig('Images/Model_Accuracy_CNN_VGG_F1_3.png')
32        fig.clf()
33        ax.cla()
34        del fig, ax
35    return

```

Source code 11: CNN code L00CV

```

1 def load_data(use_multiple_users, leave_one_out):
2     # Check if we need data from multiple users.
3     if use_multiple_users == 1:
4         range_size = 4
5         loop_start = 1
6     else:
7         range_size = 2
8         loop_start = range_size - 1
9
10    # Create empty dataframes.
11    X_all = pd.DataFrame()
12    Y_all = pd.DataFrame()
13
14    # Loop for Users.
15    for i in range(loop_start, range_size):
16        # Loop for Days.
17        for j in range(1, 4):
18            # Leave one User out for Testing.
19            if leave_one_out != i:
20                file_path = 'Data/' + 'User' + str(i) + '/Day' + str(j)
21                # Load motion data and labels.
22                print('Loading motion data for User ' + str(i) + ' Day ' + str(j) +
23                      '\n')
24
25                # Read the data.
26                x = pd.read_csv(file_path + '/Data.csv', sep=';', header=None)
27                y = pd.read_csv(file_path + '/Labels.csv', sep=';', header=None)
28
29                # Concatenate data
30                X_all = pd.concat([X_all, x], ignore_index=True, sort=False, axis=0)
31                Y_all = pd.concat([Y_all, y], ignore_index=True, sort=False, axis=0)
32                del x, y # Release memory from unused data.
33    return X_all, Y_all

```

Source code 12: CNN code L00CV

```

1 if __name__ == '__main__':
2     # To read data for all users or only 1.
3     use_multiple_users = 1
4     # Leave one out only for multiple users.
5     leave_one_out = 3
6     x_all, y_all = load_data(use_multiple_users, leave_one_out)
7     # Get number of classes and input shape.
8     classes = np.unique(y_all).size
9     input_shape = (9, 1) # For 3 sensors.
10    # Convert dataframe to numpy array.
11    x_train = x_all.to_numpy()
12    y_train = y_all.to_numpy(dtype=np.float32)
13    del x_all, y_all # Release memory from unused data.
14    # Change labels to a one-hot encoding format.
15    y_train = keras.utils.to_categorical(y_train - 1, num_classes=classes)
16
17    if use_multiple_users != 1:
18        # Split data to train test with the corresponding label.
19        x_train, x_test, y_train, y_test = train_test_split(x_train, y_train,
20                                                                test_size=0.2)
21
22    # Create the model based on the VGG-16
23    model = Sequential()
24    model.add(Conv1D(input_shape=input_shape, filters=64, kernel_size=3,
25                    padding="same", activation="relu"))
26    model.add(Conv1D(filters=32, kernel_size=3, padding="same", activation="
27                    relu"))
28    model.add(MaxPool1D(pool_size=2, strides=2))
29    model.add(Conv1D(filters=64, kernel_size=3, padding="same", activation="
30                    relu"))
31    model.add(Conv1D(filters=64, kernel_size=3, padding="same", activation="
32                    relu"))
33    model.add(MaxPool1D(pool_size=2, strides=2))
34    model.add(Flatten())
35    model.add(Dense(units=512, activation="relu"))
36    model.add(Dense(units=256, activation="relu"))
37    model.add(Dense(units=classes, activation="softmax"))

```

Source code 13: CNN code LOOCV

```

1     # Change the optimization.
2     opt = Adam(learning_rate=0.001)
3     model.compile(optimizer=opt, loss=keras.losses.categorical_crossentropy,
4                   metrics=['accuracy', keras.metrics.F1Score(average='macro'), keras.
5                       metrics.Precision(), keras.metrics.Recall()])
6
7     # Show the summary of the model.
8     model.summary()
9
10    # Set the callbacks.
11    checkpoint = ModelCheckpoint("Models/CNN3.h5", save_weights_only=True,
12                                monitor='val_accuracy', mode='max', save_best_only=True)
13    early = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=20,
14                           verbose=1, mode='auto')
15
16    # Train the model.
17    hist = model.fit(x_train, y_train, epochs=20, batch_size=10000, callbacks
18                    =[checkpoint, early], verbose=1, validation_split=0.2)
19
20    # Plot and save the Model accuracy.
21    save_plots(hist, 1)
22    save_plots(hist, 2)

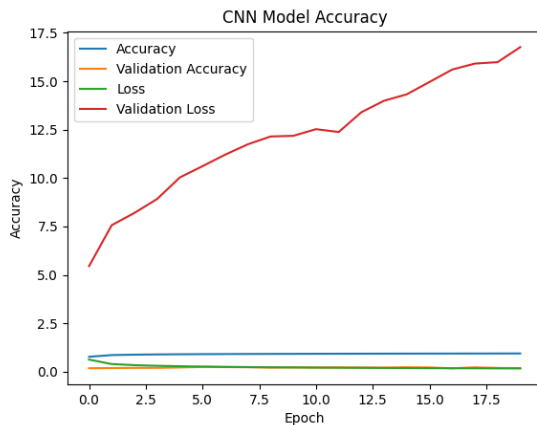
```

Source code 14: CNN code L00CV

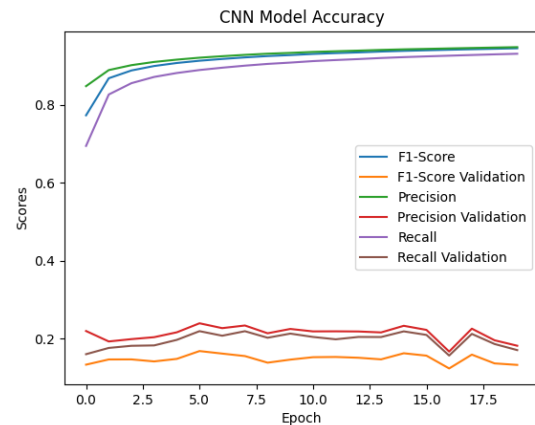
7.3 Αποτελέσματα χρηστών

7.3.1 Χρήστης 1

Εξαιτίας των λίγων εποχών, παρατηρείται ότι το μοντέλο δεν μαθαίνει αρκετά και προκύπτει μεγάλο validation loss, το οποίο αυξάνεται όσο διαρκεί η εκπαίδευση.



Σχήμα 49: Plot accuracy user1



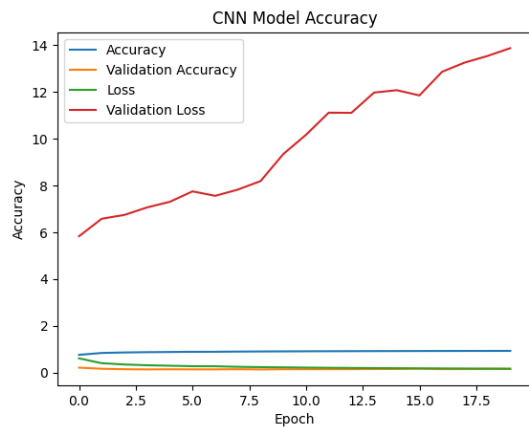
Σχήμα 50: Plot scores user1

```
Test Loss: 4.696936130523682
Test Accuracy: 0.10729824751615524
Test F1-Score: 0.06417655944824219
Test Precision: 0.10097876936197281
Test Recall: 0.09552092105150223
```

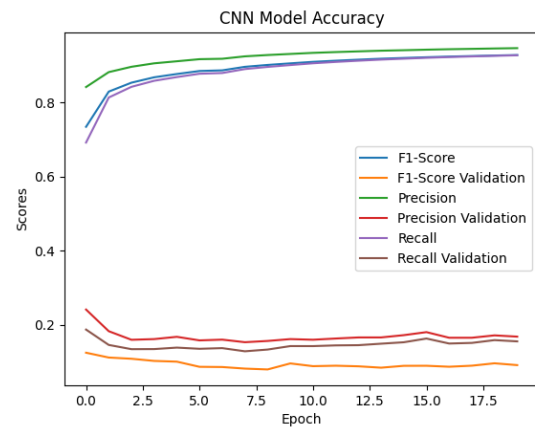
Σχήμα 51: Αποτελέσματα με τον χρήστη 1 εκτός

7.3.2 Χρήστης 2

Παρόμοια, εμφανίζεται σταδιακή αύξηση στο validation loss όπως και στον χρήστη 1. Η διαφορά τους είναι πως στην περίπτωση του χρήστη 2 δεν είναι τόσο υψηλό.



Σχήμα 52: Plot accuracy user2



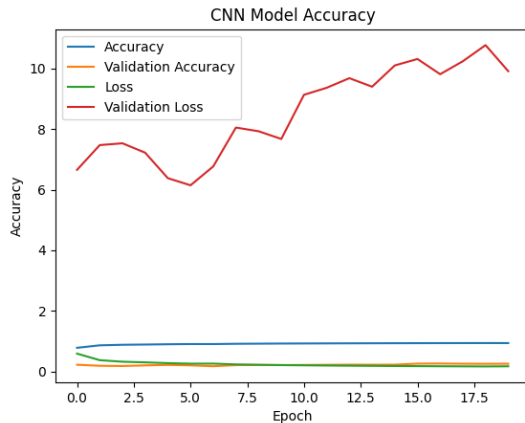
Σχήμα 53: Plot scores user2

```
Test Loss: 5.700368404388428
Test Accuracy: 0.13680943846702576
Test F1-Score: 0.08665364235639572
Test Precision: 0.1284799873828888
Test Recall: 0.10495161265134811
```

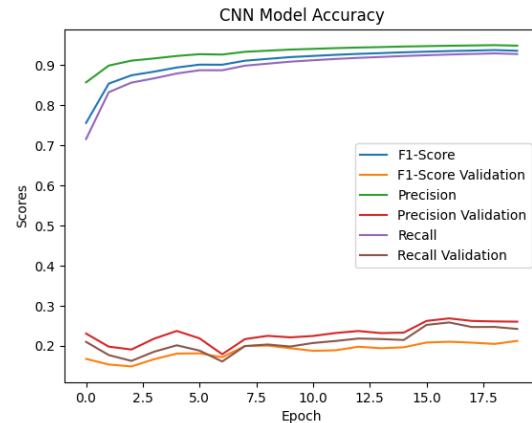
Σχήμα 54: Αποτελέσματα με τον χρήστη 2 εκτός

7.3.3 Χρήστης 3

Αντίστοιχα, γίνεται ακριβώς το ίδιο όπως και στους προηγούμενους 2 χρήστες, αλλά στον συγκεκριμένο χρήστη είναι ακόμα πιο μειωμένο το validation loss.



Σχήμα 55: Plot accuracy user3



Σχήμα 56: Plot scores user3

```
Test Loss: 10.42114485473633
Test Accuracy: 0.18247346580028534
Test F1-Score: 0.11846195161342621
Test Precision: 0.18315531313419342
Test Recall: 0.18010592460632324
```

Σχήμα 57: Αποτελέσματα με τον χρήστη 3 εκτός

7.4 Συμπέρασμα

Εξαιτίας της χρήσης λίγων εποχών, την χρήση ευστοχίας και απώλειας ταυτόχρονα στο πείραμα αλλά και λόγω της ύπαρξης μεγάλης πιθανότητας να μην υπάρχει συσχέτιση μεταξύ των χρηστών, τα αποτελέσματα που εξάγονται είναι πολύ αρνητικά. Ακόμη και στις εικόνες με τα αποτελέσματα φαίνεται πως είναι πολύ κακή τεχνική. Πιθανώς αν γίνει αλλαγή ως προς τα βάρη, δηλαδή να μην είναι μαζί η ευστοχία και η απώλεια να πάρουμε πολύ πιο ακριβή αποτελέσματα. Τέλος, μεγαλύτερος χρόνος εκπαίδευσης μπορεί να έχει καλύτερα αποτελέσματα.

8 Η χρήση του ChatGPT

Το ChatGPT[19] συνέβαλε αισθητά στην υλοποίηση του έργου, επικεντρώνοντας κυρίως στους αλγόριθμους και την επιλογή φίλτρων, καθώς και στην ανίχνευση πιθανών σφαλμάτων κατά την διάρκεια των δοκιμών. Εάν προέκυπταν ανεπίλυτα ζητήματα, έγινε αξιοποίηση της αναζήτησης στο Google, παρότι οι απαντήσεις του ChatGPT ήταν πολύ χρήσιμες και ακριβείς. Η χρήση του συνέβαλε σημαντικά στη βελτιστοποίηση των αλγορίθμων και στην αύξηση της απόδοσης.

Χωρίς αυτήν την υποστήριξη, η εργασία θα ολοκληρωνόταν, αλλά με λιγότερο ικανοποιητικά αποτελέσματα, ιδίως στα δύο πρώτα πειράματα.

9 Συμπεράσματα και μελλοντικές προεκτάσεις

Υπάρχουν αρκετοί πιθανοί τρόποι βελτίωσης, καθώς παίζουν πολλά ρόλο ώστε να βρεθεί η καλύτερη λύση. Αρχικά, θα μπορούσαν να γίνουν δοκιμές και με άλλους αλγόριθμους, πέρα από τους τρεις υπάρχοντες. Επίσης, θα μπορούσαν να χρησιμοποιηθούν διαφορετικά φίλτρα, ή περισσότερα στα ήδη υπάρχοντα, είτε να τεθούν διαφορετικές παράμετροι με σκοπό την εμφάνιση και άλλων διαφορών με τους αλγόριθμους που χρησιμοποιήθηκαν. Τα αποτελέσματα ήταν αρκετά θετικά, αλλά μπορούσαν να γίνουν και βελτιώσεις. Αντίθετα με το πρώτο και δεύτερο πείραμα, στο cross validation, θα μπορούσε να γίνει λίγο διαφορετική προσέγγιση, αφού τα αποτελέσματα είναι αρκετά χαμηλά. Βέβαια, υπάρχουν πολλοί λόγοι που μπορεί να συμβαίνει αυτό, όπως το γεγονός ότι δεν έγινε μεγάλη εκπαίδευση στο μοντέλο ή ότι ενδεχομένως τα δεδομένα να έχουν διαφορετική κατηγοριοποίηση, οπότε ίσως μπερδεύονται κάποιες κατηγορίες. Ακόμη, μπορεί να μην είναι το ιδανικό μοντέλο για το συγκεκριμένο πείραμα.

Γενικά, μπορούν να δοκιμαστούν πολλές διαφορετικές προσεγγίσεις, ώστε να βγει το καλύτερο δυνατό αποτέλεσμα. Με καλύτερα αποτελέσματα, θα υπάρχει μεγαλύτερη γνώση σχετικά με το πως θα βοηθηθεί ο κόσμος όταν θα κινείται και θα είναι πιο σωστοί οι αλγόριθμοι. Θα αναγνωρίζεται τότε κάποιος έχει πρόβλημα ή όχι, και θα βελτιωθούν πολλά προβλήματα σε αρκετά πεδία που χρησιμοποιούν αλγόριθμους για την εξαγωγή αποτελέσματος. Η μηχανική μάθηση, λοιπόν, είναι χρήσιμη σε πολλούς τομείς και παρόλο που ο συγκεκριμένος τομέας με την κίνηση είναι ένας από τους πιο δουλεμένους, δεν έχουν επιτευχθεί τα μέγιστα.

Αναφορές

- [1] David Andrés, MLPills. Regularization, October 12, 2022.
- [2] Layan Alabdullatef. Complete guide to adam optimization, September 2 2020. Accessed 28 Mar. 2023.
- [3] Mohammed Alzaylaee, Suleiman Yerima, Annette Shajan, and P. Vinod. Deep learning techniques for android botnet detection. *Electronics*, 10, 02 2021.
- [4] Gianmarco Baldini, Franc Dimc, Roman Kamnik, Gary Steri, Raimondo Giuliani, and Claudio Gentile. Identification of mobile phones using the built-in magnetometers stimulated by motion patterns. *Sensors*, 17(4):783, 2017.
- [5] Christopher Barthold, Kalyan Pathapati Subbu, and Ram Dantu. Evaluation of gyroscope-embedded mobile phones. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1632--1638, 2011.
- [6] Jason Brownlee. A gentle introduction to the rectified linear unit (relu), January 9 2019.
- [7] Fatima Es-sabery, Abdellatif Hair, Junaid Qadir, Beatriz Sainz de Abajo, Begona Garcia-Zapirain, and Isabel De la Torre Díez. Sentence-level classification using parallel fuzzy deep learning classifier. *IEEE Access*, PP:1--1, 01 2021.
- [8] James H Fetzer and James H Fetzer. *What is Artificial Intelligence?* Springer, 1990.
- [9] GeeksforGeeks. Unknown Title, Unknown Year. Accessed 28 Mar. 2023.
- [10] Davoud Gholamiangonabadi, Nikita Kiselov, and Katarina Grolinger. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. *IEEE Access*, 8:133982--133994, 2020.
- [11] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen. The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*, 6:42592--42604, 2018.
- [12] Luke Gosink, John Shalf, Kurt Stockinger, Kesheng Wu, and Wes Bethel. Hdf5-fastquery: Accelerating complex queries on hdf datasets using fast bitmap indices. In *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, pages 149--158. IEEE, 2006.
- [13] IBM. What is machine learning?, Unknown Year.
- [14] Inna Logunova. A Guide to F1 Score, July 11th, 2023.
- [15] InterviewBit. cnn-architecture, June 10, 2022.

- [16] Sajeev Kunjan, T. S. Grummett, K. J. Pope, D. M. W. Powers, S. P. Fitzgibbon, T. Bastiampillai, M. Battersby, and T. W. Lewis. The necessity of leave one subject out (loso) cross validation for eeg disease diagnosis. Berlin, Heidelberg, 2021. Springer-Verlag.
- [17] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. 12(2), 2011.
- [18] MathWorks.Inc. Gyroscope, Unknown Year.
- [19] OpenAI. GPT-3: A Language Model for Natural Language Understanding, Publication Year. Accessed Date of Access.
- [20] Charith Perera, Arkady Zaslavsky, Peter Christen, Ali Salehi, and Dimitrios Georgakopoulos. Capturing sensor data from mobile phones using global sensor networkmiddleware. 09 2012.
- [21] Shruti saxena, medium. Precision vs Recall, May 11, 2018.
- [22] Teemu Kanstrén. A Look at Precision, Recall, and F1-Score, Sep 12, 2020.
- [23] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen. Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset. *IEEE Access*, 7:10870--10891, 2019.
- [24] Lin Wang, Hristijan Gjoreski, Mathias Ciliberto, Paula Lago, Kazuya Murao, Tsuyoshi Okita, and Daniel Roggen. Three-year review of the 2018–2020 shl challenge on transportation and locomotion mode recognition from mobile sensors. *Frontiers in Computer Science*, 3, 2021.
- [25] Youming Wang, Zhao Xiao, and Gongqing Cao. A convolutional neural network method based on adam optimizer with power-exponential learning rate for bearing fault diagnosis. *Journal of Vibroengineering*, 24(4):666--678, mar 2022.
- [26] Yue Wu, Jun Li, Yu Kong, and Yun Fu. Deep convolutional neural network with independent softmax for large scale face recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1063--1067, 2016.
- [27] Yu-Dong Zhang, Chichun Pan, Junding Sun, and Chaosheng Tang. Multiple sclerosis identification by convolutional neural network with dropout and parametric relu. *Journal of computational science*, 28:1--10, 2018.
- [28] Botao Zhong, Xuejiao Xing, Peter Love, Xu Wang, and Hanbin Luo. Convolutional neural network: Deep learning-based classification of building quality problems. *Advanced Engineering Informatics*, 40:46--57, 2019.