

Αναγνώριση χρηστών με βάση τον τρόπο βαδίσματος χρησιμοποιώντας δεδομένα από επιταχυνσιόμετρο

User identification based on walking activity from accelerometer data

Ευαγγελιδάκης Λεάνδρος
Μεγάλα Δεδομένα και Αναλυτική, MSc.
Τμήμα Ψηφιακών Συστημάτων
Πανεπιστήμιο Πειραιώς
Πειραιάς, Ελλάδα
leandrosegag@hotmail.com

Abstract—Η αναγνώριση χρηστών μέσω βιομετρικών δεδομένων είναι ένα επίκαιρο πεδίο μελέτης και έχει διάφορες εφαρμογές, όπως την ασφαλή ταυτοποίηση χρηστών, την εξατομίκευση κ.λπ. Οι συνηθέστεροι βιομετρικοί δείκτες που χρησιμοποιούνται σήμερα είναι τα δακτυλικά αποτυπώματα, η ίριδα του ματιού και τα χαρακτηριστικά του προσώπου και της φωνής. Πλέον, όλες σχεδόν οι ηλεκτρονικές συσκευές διαθέτουν κάποιον αισθητήρα κίνησης, όπως επιταχυνσιόμετρα, γυροσκόπια κ.λπ. και έτσι καθίσταται εύκολη η διαδικασία καταγραφής κινησιολογικών βιομετρικών δεδομένων. Στην παρούσα εργασία, σκοπός είναι η αναγνώριση ενός χρήστη, με βάση τον τρόπο βαδίσματος, από δεδομένα που έχουν συλλεχθεί χρησιμοποιώντας το επιταχυνσιόμετρο Android κινητών τηλεφώνων. Όπως έχει αποδειχθεί, ο τρόπος βαδίσματος μπορεί να αποτελέσει επαρκή δείκτη για την αναγνώριση του χρήστη λόγω της ύπαρξης στοιχείων μοναδικότητας από άνθρωπο σε άνθρωπο. Ως επέκταση της δουλειάς προηγούμενων δημοσιεύσεων για τα δεδομένα που αναλύουμε, κάνουμε μια πιο προσεκτική επιλογή παραμέτρων για διαφόρους αλγόριθμους μηχανικής μάθησης που έχουν χρησιμοποιηθεί, πετυχαίνοντας υψηλότερα αποτελέσματα και εξετάζουμε επιπλέον κάποια πιο σύνθετα μοντέλα.

Λέξεις κλειδιά—επιβλεπόμενη μηχανική μάθηση, κατηγοριοποίηση, ταυτοποίηση χρηστών, τρόπος βαδίσματος, επιταχυνσιόμετρο

I. ΕΙΣΑΓΩΓΗ

Η ψηφιακή ταυτοποίηση χρηστών στις μέρες μας χρησιμοποιεί κυρίως βιομετρικά δεδομένα όπως το δακτυλικό αποτύπωμα, την ίριδα του ματιού, τη φωνή κ.λπ. Η μελέτη δεδομένων από κάποιου είδους φυσική δραστηριότητα, όπως όταν κάποιος περπατάει ή τρέχει, έχει αναδειχθεί πρόσφατα σε ιδιαίτερα ενδιαφέρον ζήτημα είτε για την αναγνώριση του χρήστη που εμπλέκεται στην εκάστοτε δραστηριότητα είτε για την αναγνώριση του είδους της δραστηριότητας.

Ο τρόπος βαδίσματος (gait) έχει ιδιαίτερο ενδιαφέρον καθώς, προφανώς, είναι η πιο συνήθης ανθρώπινη φυσική δραστηριότητα αλλά και καθώς διάφορες μελέτες έχουν δείξει ότι αυτός διαφέρει, σε ένα βαθμό, από άνθρωπο σε άνθρωπο. Η ανάλυση αυτού, έχει εξεταστεί στο παρελθόν,

αρχικά ως πρόβλημα αναγνώρισης εικόνας [1],[2], χρησιμοποιώντας τα οπτικά δεδομένα από κάμερες, και στη συνέχεια χρησιμοποιώντας δεδομένα από αισθητήρες όπως επιταχυνσιόμετρα [3] και γυροσκόπια [4]. Στη δημοσίευση [5] έχει γίνει μια επισκόπηση της έως τότε υπάρχουσας δουλειάς και έχουν χρησιμοποιηθεί έξι αλγόριθμοι, πετυχαίνοντας υψηλή ακρίβεια (94%) κατηγοριοποίησης χρηστών, για δεδομένα από επιταχυνσιόμετρα. Στην παρούσα εργασία εξετάζουμε το ίδιο σύνολο δεδομένων του [5]. Ως επέκταση της υπάρχουσας δουλειάς, κάνουμε μια προσεκτικότερη επιλογή παραμέτρων για τους αλγόριθμους που έχουν χρησιμοποιηθεί, πετυχαίνοντας υψηλότερα αποτελέσματα, και επιπλέον εφαρμόζουμε νευρωνικά δίκτυα τα οποία φαίνεται να αποδίδουν ελαφρώς καλύτερα συγκριτικά με τους κλασσικούς αλγόριθμους.

Τα δεδομένα [6] που εξετάζουμε, αφορούν 22 συμμετέχοντες οι οποίοι διένυσαν μια διαδρομή στην ύπαιθρο και οι μετρήσεις της κινήσεώς τους καταγράφηκαν από το επιταχυνσιόμετρο Android τηλεφώνων. Το επιταχυνσιόμετρο καταγράφει την επιτάχυνση για τους τρεις άξονες x,y και z ανά κάποιο χρονικό διάστημα t.

II. ΜΕΘΟΔΟΛΟΓΙΑ

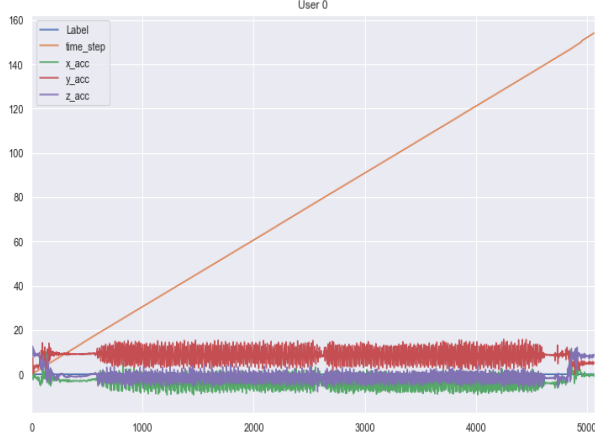
Η μεθοδολογία που ακολουθούμε για την επίλυση του προβλήματος είναι η εξής:

- Kαταγραφή της επιτάχυνσης ως προς τους τρεις άξονες μέσω επιταχυνσιόμετρου.
- Ανάλυση των δεδομένων και εξαγωγή των κατάλληλων χαρακτηριστικών από αυτά.
- Εκπαίδευση αλγορίθμων μηχανικής μάθησης σε ένα ποσοστό των δεδομένων και εφαρμογή στο υπόλοιπο.
- Αξιολόγηση της αποτελεσματικότητας των μοντέλων μέσω μετρικών.

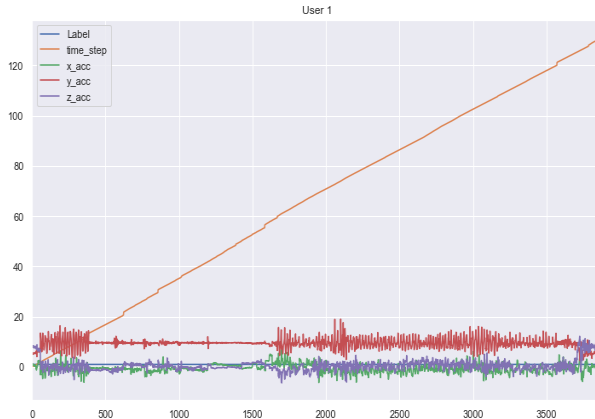
1. Εξαγωγή χαρακτηριστικών

Όπως αναφέρθηκε, τα δεδομένα αποτελούνται από τέσσερις μεταβλητές. Το χρονικό στιγμιότυπο (timestamp) της κάθε μέτρησης και την επιτάχυνση ως προς τους τρεις άξονες x, y και z.

Στις Εικόνες 1 και 2, φαίνονται τα δεδομένα για τους δύο πρώτους χρήστες. Όπως παρατηρείται εκ πρώτης όψεως, υπάρχει σε ένα βαθμό διαφορά μεταξύ τους.



Εικόνα 1. Δεδομένα από επιταχυνσιόμετρο (t, x, y, z). Χρήστης 0



Εικόνα 2. Δεδομένα από επιταχυνσιόμετρο (t, x, y, z). Χρήστης 1

Από προηγούμενες δημοσιεύσεις, όπως του [5], έχουν αναφερθεί εκτενώς χαρακτηριστικά που φαίνεται να ερμηνεύουν τα δεδομένα όσο το δυνατόν καλύτερα και αυτά χρησιμοποιούμε και εμείς. Αρχικά, προσθέτουμε έναν επιπλέον άξονα ο οποίος είναι το μέτρο του διανύσματος που προκύπτει με συντεταγμένες τις επιταχύνσεις των τριών αξόνων. Αυτός είναι η ποσότητα $R = \sqrt{a_x^2 + a_y^2 + a_z^2}$.

Έτσι, τα δεδομένα αποτελούνται από τέσσερις άξονες. Για κάθε έναν από αυτούς, εξάγουμε 12 χαρακτηριστικά τα οποία φαίνονται στον παρακάτω πίνακα.

Χαρακτηριστικό	Περιγραφή
Mean	Means
Std	Standard deviation
Var	Variance

Min	Minimum value
Max	Maximum value
Acf_mean	Auto correlation mean
Acf_std	Auto correlation standard deviation
Acv_mean	Auto covariance mean
Acv_std	Auto covariance standard deviation
Skew	Skewness
Kurtosis	Kurtosis
error	Deviation from mean

Πίνακας 1. Εξαγωγή χαρακτηριστικών

Συνεπώς, το τελικό σύνολο δεδομένων που αναλύουμε περιέχει συνολικά 48 χαρακτηριστικά, 12 για κάθε άξονα. Για την εξαγωγή των χαρακτηριστικών αυτών, τα δεδομένα έχουν χωριστεί ανά κάποιο χρονικό διάστημα (time-window), ώστε να έχουμε μια σχετικώς ομοιόμορφη καταμέριση των δεδομένων κάθε χρήστη. Συγκεκριμένα, επιλέγονται 100 καταχωρήσεις με 50% επικάλυψη μεταξύ τους, για κάθε άξονα.

2. Εφαρμογή αλγορίθμων μηχανικής μάθησης

Το επόμενο βήμα μετά την εξαγωγή των χαρακτηριστικών που θα χρησιμοποιήσουμε, είναι η εφαρμογή σε αυτά αλγορίθμων μηχανικής μάθησης για την κατηγοριοποίηση των χρηστών. Χρησιμοποιούμε διάφορους από τους δημοφιλέστερους «κλασσικούς» αλγορίθμους (SVM, KNN, RF) που έχουν χρησιμοποιηθεί και στο [5] αλλά και πιο σύνθετα μοντέλα όπως Νευρωνικά Δίκτυα.

Για κάθε αλγόριθμο εξετάζουμε διάφορες τιμές παραμέτρων και παραθέτουμε τα καλύτερα αποτελέσματα. Η εξέταση αυτή έγινε μέσω Grid Search CV, δηλαδή, θέτουμε ένα εύρος τιμών για κάθε παράμετρο και κάνουμε εξαντλητική εφαρμογή σε όλους τους δυνατούς συνδυασμούς μεταξύ αυτών, υπολογίζοντας την βέλτιστη επιλογή παραμέτρων. Φυσικά αυτή η διαδικασία χρησιμοποιήθηκε για μικρά σχετικά εύρη παραμέτρων καθότι είναι μια αρκετά χρονοβόρα διαδικασία.

3. Αξιολόγηση

Για την αξιολόγηση των αλγορίθμων χρησιμοποιούμε 10-fold Cross Validation και έτσι υπολογίζουμε τη μέση ακρίβεια και το αντίστοιχο σφάλμα. Επιπλέον, υπολογίζουμε τα Accuracy, Recall και F-score για κάθε κατηγορία, δηλαδή για κάθε χρήστη, για τα οποία φτιάχνουμε και τα αντίστοιχα διαγράμματα.

III. ΑΠΟΤΕΛΕΣΜΑΤΑ

Το αρχικό σύνολο δεδομένων βρίσκεται στην ιστοσελίδα του UCI Machine learning repository [6] και περιέχει ένα αρχείο csv για κάθε χρήστη, όπου βρίσκονται οι μετρήσεις από το επιταχυνσιόμετρο. Η εξαγωγή των χαρακτηριστικών έχει ήδη γίνει από προηγούμενες δημοσιεύσεις και τα δεδομένα βρίσκονται διαθέσιμα στο [7].

Για την επεξεργασία των δεδομένων, χρησιμοποιούμε τη γλώσσα προγραμματισμού Python, χρησιμοποιώντας τις βιβλιοθήκες scikit-learn και tensorflow για το κομμάτι της μηχανικής μάθησης.

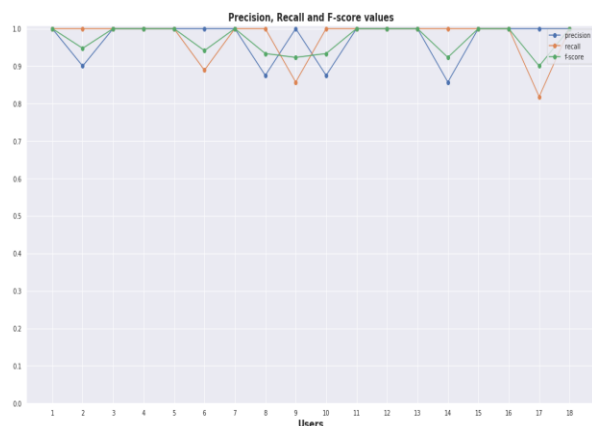
Το σύνολο των δεδομένων που εξετάζουμε μετά την εξαγωγή χαρακτηριστικών, αποτελείται από 393 καταχωρήσεις, με 49 μεταβλητές, όπου η πρώτη είναι η ετικέτα του χρήστη και οι υπόλοιπες 48 τα χαρακτηριστικά που θα αναλύσουμε. Επιλέγουμε το 60% των δεδομένων για εκπαίδευση και το υπόλοιπο 40% για επαλήθευση.

1. Support Vector Machines (SVM)

Οι Μηχανές Διανυσμάτων Υποστήριξης (SVM) αναζητούν το βέλτιστο υπερεπίπεδο που διαχωρίζει τα δεδομένα.

Για τις παραμέτρους του αλγορίθμου χρησιμοποιούμε τη γραμμική πυρήνα με παράμετρο κανονικοποίησης ίση με 1 (kernel = 'linear', C=1).

Η ακρίβεια από το 10-fold Cross Validation είναι: 0.95 (+/- 0.09).

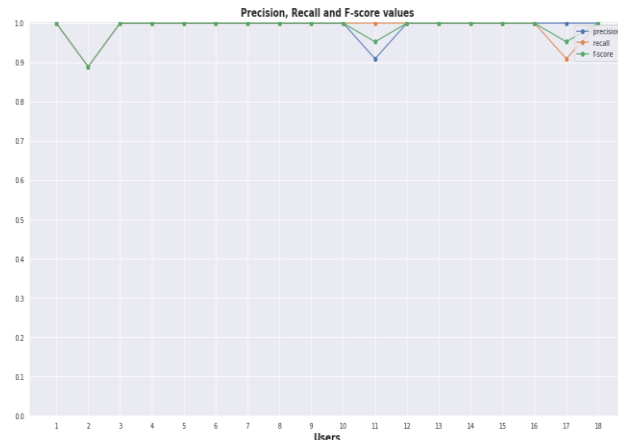


Εικόνα 3. Support Vector Machines (kernel='linear', C=1)

Στην Εικόνα 3 βλέπουμε ότι όλες οι μετρικές (Precision, Recall, F-score) έχουν τιμή πάνω από 0.8 για κάθε χρήστη. Για τους περισσότερους χρήστες, πετυχαίνουμε απόλυτη επιτυχία στην κατηγοριοποίηση.

Επίσης, εξετάζουμε και την περίπτωση όπου kernel='rbf' (Radial basis function). Για την παράμετρο γ καταλήγουμε στο 0.02 μετά από δοκιμές (gamma=0.02).

Η ακρίβεια από το 10-fold Cross Validation είναι: 0.95 (+/- 0.06).



Εικόνα 4. Support Vector Machines (kernel='rbf', gamma=0.02)

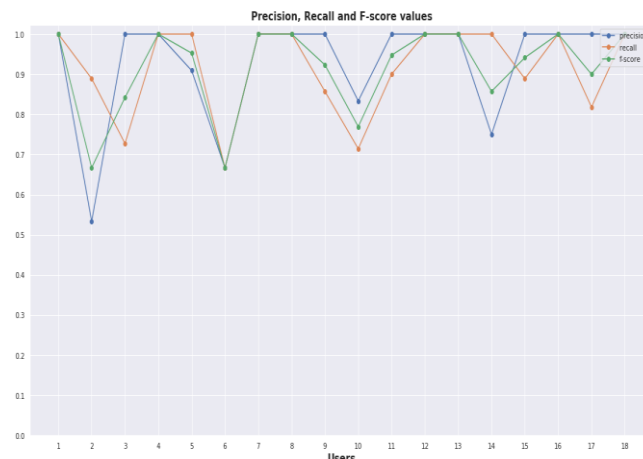
Στην Εικόνα 4 παρατηρούμε ότι έχουμε ιδιαίτερα υψηλά αποτελέσματα σχεδόν για όλους τους χρήστες. Συγκεκριμένα έχουμε πλήρη επιτυχία κατηγοριοποίησης όλων των χρηστών, εκτός από τους χρήστες 2, 11 και 17. Βέβαια, ακόμη και για αυτούς, όλες οι μετρικές σχεδόν ξεπερνούν το 0.9.

2. Stochastic Gradient Decent (SGD)

Η μέθοδος της SGD χρησιμοποιείται σε διάφορα γραμμικά μοντέλα για τον προσδιορισμό του ρυθμού εκμάθησης (learning rate), εξετάζοντας την παράγωγο της loss function σε κάθε βήμα και ενημερώνοντας το μοντέλο κατά τη διάρκεια της εκπαίδευσης.

Για την εφαρμογή SGD χρησιμοποιούμε ως loss function τη μετρική 'hinge' με παράμετρο $\alpha = 0.001$, δηλαδή πρόκειται για SVM κατηγοριοποιητή. Ως ποινή (penalty) χρησιμοποιούμε τη νόρμα L1 με συντελεστή 0.5. Συνολικά έχουμε: SGDClassifier (alpha=0.001, loss='hinge', penalty='l1', l1_ratio=0.5).

Η ακρίβεια από το 10-fold Cross Validation είναι: 0.90 (+/- 0.10).



Εικόνα 5. SGDClassifier (alpha= 0.001, l1_ratio= 0.5, loss= 'hinge', penalty= 'l1')

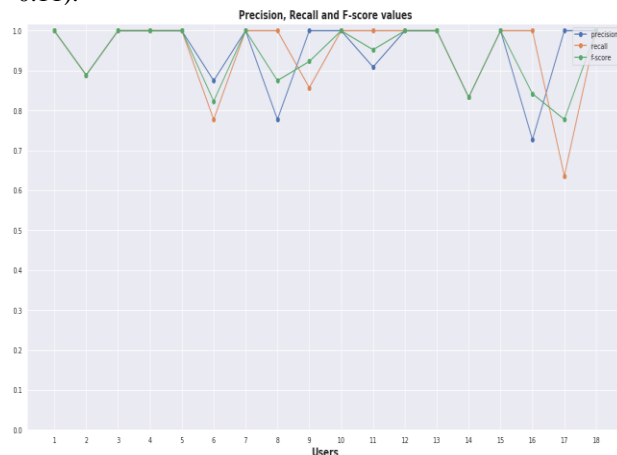
Όπως φαίνεται από τα αποτελέσματα στην Εικόνα 5, η εφαρμογή SGD στα SVM δεν φαίνεται να πετυχαίνει καλύτερα αποτελέσματα. Φυσικά, και εδώ ορισμένοι χρήστες κατηγοριοποιούνται σωστά με μεγάλη επιτυχία.

3. K-Nearest Neighbors

Ο κατηγοριοποιητής των K-Πλησιέστερων Γειτόνων επιλέγει την κλάση εξετάζοντας τα K πλησιέστερα σημεία, πλειοψηφικά. Επίσης, στα σημεία μπορούν να ανατεθούν βάρη με διάφορες τεχνικές, ώστε κάποια από αυτά να συνεισφέρουν περισσότερο ή λιγότερο.

Για την εφαρμογή του αλγορίθμου των K-Πλησιέστερων Γειτόνων, επιλέγουμε K=5 γείτονες, χρησιμοποιώντας την Ευκλείδεια απόσταση (Minkowski, p=2) και επιλέγοντας ομοιόμορφη κατανομή βαρών για τα σημεία.

Η ακρίβεια από το 10-fold Cross Validation είναι: 0.93 (+/- 0.11).



Εικόνα 6. KNeighborsClassifier ($n_neighbors=5$, $weights = 'uniform'$, $p = 2$, $metric = 'minkowski'$)

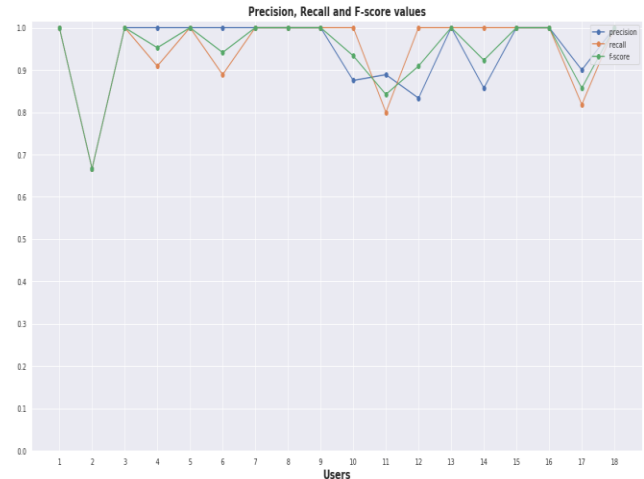
Όπως φαίνεται στην Εικόνα 6, ο αλγόριθμος αποδίδει ικανοποιητικά και γενικά τα αποτελέσματα βρίσκονται πάνω από το 0.7 ενώ και εδώ κάποιοι χρήστες κατηγοριοποιούνται με απόλυτη επιτυχία.

4. Random Forests

Ο ταξινομητής Τυχαίων Δένδρων χρησιμοποιεί πολλά ανεξάρτητα δένδρα για την τελική εκτίμηση (bagging) και ελαχιστοποιεί το overfitting. Για τον κατηγοριοποιητή επιλέγουμε της παρακάτω παραμέτρους:

- Πλήθος εκτιμητών ($n_estimators$) = 150,
- Μέγιστο βάθος (max_depth) = 120 και
- Μέγιστο πλήθος χαρακτηριστικών ($max_features$) = 30.

Η ακρίβεια από το 10-fold Cross Validation είναι: 0.94 (+/- 0.11).



Εικόνα 7. RandomForestClassifier ($max_depth=120$, $n_estimators = 150$, $max_features=30$)

Με εξαίρεση τον χρήστη 2, για όλους τους υπόλοιπους ο αλγόριθμος πετυχαίνει σκορ τουλάχιστον 0.8 και γενικά παρουσιάζει ικανοποιητικά αποτελέσματα.

5. Neural Networks

Η χρήση νευρωνικών δικτύων όπως φάνηκε, παρουσιάζει τα καλύτερα αποτελέσματα, με κόστος τον παραπάνω χρόνο που απαιτεί για την εκπαίδευση. Για την εύρεση των κατάλληλων παραμέτρων χρησιμοποιήσαμε Grid Search, κατά την οποία εξετάσαμε διαφορετικές τιμές για τις επιμέρους παραμέτρους. Οι διάφορες επιλογές που εξετάσαμε για κάθε μία φαίνονται παρακάτω:

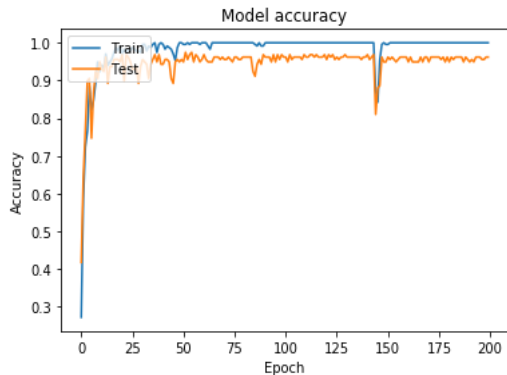
- #Hidden layers: [1, 2, 3, 4, 5]
- #Layer size: [5, 10, 20, 48, 2*48, 5*48, 20*48]
- Activation Function: ['relu', 'sigmoid', 'tanh', 'softmax', 'exponential', 'linear']
- Optimizer: ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']
- Dropout : [0.0, 0.1, 0.5, 0.8]
- Batch size : [6,12]
- #Epochs : [100,200,300]

Τελικά, καταλήγουμε στο εξής μοντέλο:

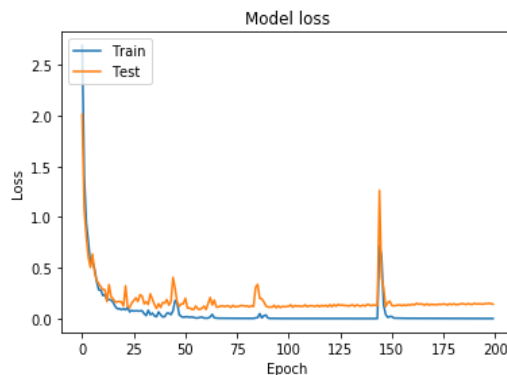
Αρχικά παίρνουμε ένα Dense layer που δέχεται ως είσοδο τα δεδομένα (48 διαστάσεις) με συνάρτηση ενεργοποίησης την 'relu' (Rectified Linear). Στη συνέχεια προσθέτουμε ένα δεύτερο Dense layer διάστασης 18, το οποίο ενεργοποιείται με τη συνάρτηση 'softmax'. Τέλος, χρησιμοποιούμε ως μετρική σφάλματος (loss function) την 'Categorical Crossentropy', όπως συνίσταται σε περιπτώσεις multiclass classification, και με βελτιστοποιητή (optimizer) τον 'Adam'. Τα δεδομένα διαβάζονται με batch size = 6 και το μοντέλο εκπαιδεύεται για 200 epochs. Εξετάσαμε και την προσθήκη Dropout layers, τα οποία χρησιμοποιούνται για τον περιορισμό του overfit,

όμως από τα αποτελέσματα των Grid Search καταλήξαμε στο ότι κάτι τέτοιο δεν προσφέρει καλύτερα αποτελέσματα για τα συγκεκριμένα δεδομένα.

Η ακρίβεια από το 10-fold Cross Validation είναι: 0.96 (+/- 0.05).



Εικόνα 8. Ακρίβεια μοντέλου



Εικόνα 9. Σφάλμα μοντέλου

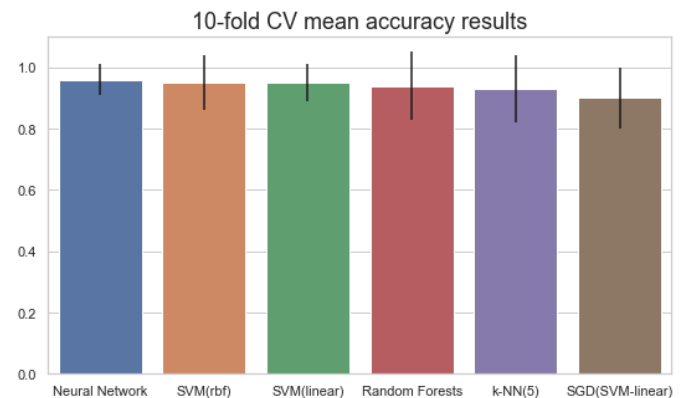
Στις Εικόνες 8 και 9, φαίνεται η μεταβολή της ακρίβειας και του σφάλματος κατά τη διαδικασία της εκπαίδευσης. Στο τελικό βήμα της εκπαίδευσης, ο αλγόριθμος επιτυγχάνει Test (validation) accuracy: 0.96 και Validation loss: 0.14.

6. Σύγκριση αποτελεσμάτων

Στον Πίνακα 2 φαίνονται τα αποτελέσματα των 10-fold Cross Validation για τους αλγόριθμους που χρησιμοποιήθηκαν.

Αλγόριθμος	Ακρίβεια	Απόκλιση
SVM(linear)	0.95	0.09
SVM(rbf)	0.95	0.06
SGD(SVM-linear)	0.90	0.10
K-NN(5)	0.93	0.11
Random Forests	0.94	0.11
Neural Network	0.96	0.05

Πίνακας 2. Αποτελέσματα 10-fold CV



Εικόνα 10. Ραβδόγραμμα με τα αποτελέσματα των 10-fold CV

Όπως φαίνεται από τα αποτελέσματα, οι καλύτερες επιδόσεις επιτυγχάνονται από Νευρωνικά Δίκτυα και αμέσως μετά από τις Μηχανές Διανυσμάτων Υποστήριξης (SVM).

IV. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα εργασία μελετήσαμε το πρόβλημα της αναγνώρισης χρηστών με βάση τον τρόπο βαδίσματος, χρησιμοποιώντας δεδομένα από επιταχυνσιόμετρο Android τηλεφώνων. Όπως φαίνεται, ο τρόπος βαδίσματος εμπεριέχει πράγματι στοιχεία μοναδικότητας και μπορεί να γίνει αρκετά καλή αναγνώριση ενός χρήστη με βάση αυτόν.

Για την πειραματική διαδικασία, εξετάσαμε διάφορους αλγορίθμους κατηγοριοποίησης τους οποίους αξιολογήσαμε μέσω 10-fold Cross Validation και εξετάζοντας την απόδοσή τους για όλους τους χρήστες ξεχωριστά. Η εργασία βασίστηκε σε προηγούμενες δημοσιεύσεις για τα συγκεκριμένα δεδομένα. Όπως δείξαμε, με την προσεκτική επιλογή παραμέτρων για τους αλγορίθμους, μπορούμε να πετύχουμε εξαιρετικά καλά αποτελέσματα ακόμη και με απλούς σχετικά αλγορίθμους. Τα καλύτερα αποτελέσματα σε ότι αφορά τους κλασσικούς αλγορίθμους μηχανικής μάθησης προκύπτουν από τις Μηχανές Υποστήριξης Διανυσμάτων (SVM(kernel=’ rbf’)) , επιτυγχάνοντας μέση ακρίβεια (10-fold CV): 95% (+/- 6%). Επιπλέον, χρησιμοποιώντας Νευρωνικά Δίκτυα, δείξαμε ότι μπορούμε να επιτύχουμε τουλάχιστον ίση ή και καλύτερη ακρίβεια, συγκριτικά με όλους τους άλλους αλγορίθμους που είδαμε. Συγκεκριμένα, κατασκευάσαμε μοντέλο με ακρίβεια (10-fold CV): 96% (+/- 5%).

Όπως είδαμε, όλοι σχεδόν οι αλγόριθμοι κατηγοριοποιούν κάποιους χρήστες καλύτερα απ’ότι κάποιους άλλους. Το γεγονός αυτό, σε συνδυασμό με το γεγονός ότι για όλους τους αλγορίθμους πετυχαίνουμε υψηλής ακρίβειας προβλέψεις, θα έδινε ενδιαφέρον, μελλοντικά, στη συνδυαστική χρήση διαφόρων κατηγοριοποιητών, όπως όταν εφαρμόζουμε τεχνικές τύπου Voting ή Stacking.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Lily Lee and Eric Grimson. Gait analysis for recognition and classification. In IEEE Conference on Face and Gesture Recognition, pages 155–161, 2002 J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, pp.68–73, 1892,
- [2] Lily Lee and Eric Grison. Gait Appearance for Recognition. Lecture Notes in Computer Science, 2359:143– 154, 2002.
- [3] Casale, P. Pujol, O. and Radeva, P. Personalization and user verification in wearable systems using biometric walking patterns. Personal and Ubiquitous Computing, 16(5), 563-580, 2012.] M. F. Nowlan, "Human Identification via Gait Recognition Using Accelerometer Gyro Forces," 2009.
- [5] M. F. Nowlan, "Human Identification via Gait Recognition Using Accelerometer Gyro Forces," 2009.
- [6] Umair Ahmed "User identification and classification using raw accelerometer data from walking activity" 2016
- [7] <http://archive.ics.uci.edu/ml/datasets/User+Identification+From+Walking+Activity#>