

ΑΥΤΟΜΑΤΗ ΚΑΤΑΤΜΗΣΗ ΚΑΙ ΚΑΤΗΓΟΡΙΟΠΟΙΣΗ
ΤΗΣ ΠΟΛΥΔΙΑΣΤΑΤΗΣ ΧΡΟΝΟΣΕΙΡΑΣ
ΑΙΣΘΗΤΗΡΙΑΚΩΝ ΣΗΜΑΤΩΝ

Γεώργιος Βαρδάκας



Επιστημονικός υπεύθυνος έργου:
Στέργιος Αναστασιάδης (Αναπληρωτής Καθηγητής)

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πολυτεχνική Σχολή Πανεπιστημίου Ιωαννίνων

Περιεχόμενα

1	Εισαγωγή	2
2	Συλλογή Δεδομένων	2
3	Αυτόματη Κατάτμηση	3
4	Εξαγωγή Χαρακτηριστικών	4
5	Προ-επεξεργασία Χαρακτηριστικών	4
6	Μείωση της διάστασης	5
7	Το μοντέλο	5
8	Βελτιστοποίηση των υπερπαραμέτρων	6
9	Αξιολόγηση	7
10	Πειραματικά αποτελέσματα	8
11	Βιβλιοθήκες που χρησιμοποιήθηκαν	8

1 Εισαγωγή

Στην παρούσα εργασία κληθήκαμε να λύσουμε το πρόβλημα της αυτόματης κατάτμησης και κατηγοριοποίησης της πολυδιάστατης χρονοσειράς αισθητηριακών σημάτων στα πλαίσια του έργου **ΉΜΟΜΟΡΕ:ΕΞΥΠΝΟ ΣΥΣΤΗΜΑ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΤΗΣ ΦΥΣΙΟΛΟΓΙΚΗΣ ΔΙΑΒΙΩΣΗΣ ΗΛΙΚΙΩΜΕΝΩΝ ΣΕ ΑΣΤΙΚΕΣ ΚΑΙ ΑΓΡΟΤΙΚΕΣ ΠΕΡΙΟΧΕΣ** με κωδικό **82475**. Για να το πετύχουμε αυτό αρχικά συλλέχθηκαν τα δεδομένα που είναι απαραίτητα για την κατασκευή ενός μαθηματικού μοντέλου μηχανικής μάθησης, και στην συνέχεια ακολούθησε η προ-επεξεργασία τους, ώστε να είναι κατάλληλα προετοιμασμένα για να εισαχθούν στο μοντέλο. Επιπρόσθετα μελετήθηκαν διάφορα μοντέλα μηχανικής μάθησης, μέχρι να βρεθεί αυτό με τα καλύτερα αποτελέσματα. Το τελικό μοντέλο περιέχει εσωτερικά τα βήματα της προ-επεξεργασίας αλλά και της πρόβλεψης με αυτόματο τρόπο και βασίστηκε σε μεγάλο βαθμό στην μεθοδολογία [1]. Τελικά η αξιολόγηση του μοντέλου μηχανικής μάθησης για τον έλεγχο της ποιότητας των αποτελεσμάτων του πραγματοποιήθηκε με (group) K-fold cross validation.

2 Συλλογή Δεδομένων

Μία προαπαιτήση για την κατασκευή ενός μαθηματικού μοντέλου μηχανικής μάθησης είναι φυσικά τα δεδομένα, καθώς αυτές οι μέθοδοι μαθαίνουν μέσα από αυτά. Για την συλλογή των δεδομένων έγινε χρήση του έξυπνου ρολογιού Fitbit Versa. Το έξυπνο αυτό ρολόι, μας δίνει την δυνατότητα να αντλήσουμε τις μετρήσεις που καταγράφουν οι διαθέσιμοι αισθητήρες του. Οι αισθητήρες αυτοί αποτελούνται από από το γυροσκόπιο, το επιταχυνσιόμετρο καθώς και τον αισθητήρα καρδιακών παλμών. Οι αισθητήρες του γυροσκοπίου και του επιταχυνσιόμετρου καταγράφουν μετρήσεις με συχνότητα 10 Η ενώ ο αισθητήρας καρδιακών παλμών καταγράφει μετρήσεις με συχνότητα 1 Η. Στην παρούσα εργασία έγινε χρήση μόνο των αισθητήρων του γυροσκοπίου και του επιταχυνσιόμετρου καθώς οι καταγραφές του αισθητήρα των καρδιακών παλμών δεν περιείχε χρήσιμη πληροφορία για την αυτόματη κατηγοριοποίηση της πολυδιάστατης χρονοσειράς των αισθητήρων. Συνολικά καταφέραμε να συλλέξουμε δεδομένα από επτά διαφορετικούς χρήστες. Τα δεδομένα είναι μορφής πίνακα όπως ο παρακάτω:

	TIMESTAMP	ACCEL_X	ACCEL_Y	ACCEL_Z	GYRO_X	GYRO_Y	GYRO_Z	ACTIVITY_ID	USER_ID
0	2020-08-25 16:23:14.590	-1.230657	5.430215	7.670058	-1.084472	0.181100	-0.076701	107	0
1	2020-08-25 16:23:14.690	-1.743032	4.716721	8.117788	-1.465848	0.301478	-0.136357	107	0
2	2020-08-25 16:23:14.790	-2.229070	4.455745	8.960572	-1.877052	0.394159	-0.137423	107	0
3	2020-08-25 16:23:14.890	-2.496031	3.766193	9.097046	-1.774784	0.359005	-0.070309	107	0
4	2020-08-25 16:23:14.990	-2.711515	2.225478	9.026415	-2.076263	0.344090	-0.100138	107	0

Σχήμα 1: Πρώτες πέντε εγγραφές του πίνακα δεδομένων.

όπου η κολόνα `TIMESTAMP` αναφέρεται στο χρόνο δειγματοληψίας της εγγραφής, η κολόνα `ACCEL_{X,Y,Z}` αναφέρεται στην μέτρηση του επιταχυνσιομέτρου στον άξονα $\{x, y, z\}$, η κολόνα `GYRO_{X,Y,Z}` αναφέρεται στην μέτρηση του γυροσκοπίου στον άξονα $\{x, y, z\}$, η κολόνα `ACTIVITY_ID` δηλώνει την δραστηριότητα που πραγματοποιεί ο χρήστης και τέλος η κολόνα `USER_ID` αναφέρεται στον ποίος χρήστης έκανε την δραστηριότητα. Οι μετρήσεις των αισθητήρων καταγράφουν δείγματα στον τρισδιάστατο χώρο, αυτός είναι και ο λόγος που κάθε αισθητήρας έχει μετρήσεις τριών τυχαίων μεταβλητών (x, y, z) . Επίσης ο κάθε χρήστης έχει μοναδικό ID. Πλέον τα δεδομένα έχουν κατασκευαστεί και αποθηκευτεί με τέτοιο τρόπο, ώστε να είναι έτοιμα για να χρησιμοποιηθούν για το πρόβλημα της αυτόματης ταξινόμησης ακολουθίας $\{(X_i, y_i)\}_{i=1}^N$, με κατηγορία y_i για κάθε ακολουθία X_i . Η κάθε ακολουθία X_i μοντελοποιείται σαν πολυδιάστατη χρονοσειρά αισθητηριακών σημάτων, έχοντας T_i δείγματα $\langle x_1, x_2, \dots, x_{T_i} \rangle_i$, με κατηγορία δραστηριότητας y_i . Τέλος το κάθε δείγμα $x_j = [ACCEL_X, ACCEL_Y, ACCEL_Z, GYRO_X, GYRO_Y, GYRO_Z]_j$ και με $y_j = ACTIVITY_ID$. Για την διαχείριση των δεδομένων έγινε χρήση της βιβλιοθήκης `pandas` [2].

3 Αυτόματη Κατάτμηση

Μετά την συλλογή των δεδομένων που μοντελοποιούνται σαν πολυδιάστατη χρονοσειρά αισθητηριακών σημάτων ακολουθεί η προ-επεξεργασία τους. Το πρώτο βασικό βήμα της προ-επεξεργασίας είναι η αυτόματη κατάτμηση του σήματος. Για την επίτευξη της αυτόματης κατάτμησης έγινε χρήση της συνάρτησης `Segment(width, overlap)` της βιβλιοθήκης `seglearn` [3]. Η συνάρτηση λαμβάνει ως είσοδο το αρχικό σήμα και το τμηματοποιεί σε τμήματα μεγέθους `width`, κάνοντας χρήση ενός επικαλυπτόμενου κυλίστρου (sliding

window) σταθερού μήκους, στην περίπτωση μας το width ίσο με 5 δευτερόλεπτα. Με αυτήν την μέθοδο κατασκευάζουμε ένα τρισδιάστατο χρονικό τένσορα $\phi_i = \langle W_1, \dots, W_M \rangle$ και για κάθε χρονοσειρά X_i . Ο τένσορας ϕ_i έχει σχήμα $(M_i, width, 6)$ με $width$ το μήκος του παραθύρου και M_i ο αριθμός των παραθύρων που κατασκευάστηκαν για κάθε χρονοσειρά X_i . Το τελικό σύνολο δεδομένων είναι το σύνολο όλων των τμημάτων που παρήχθησαν από το κυλιόμενο παράθυρο και συμβολίζεται $\{W_i, y_i\}_{i=1}^{N_w}$, όπου το N_w είναι το πλήθος των τμημάτων του συνόλου δεδομένων. Το πρόβλημα μηχανικής μάθησης της ταξινόμησης των πολυδιάστατων χρονοσειρών του έργου πλέον διατυπώθηκε και αξιολογήθηκε ως ταξινόμηση του τμηματοποιημένου συνόλου δεδομένων $\{W_i, y_i\}_{i=1}^{N_w}$.

4 Εξαγωγή Χαρακτηριστικών

Το επόμενο βήμα είναι η εξαγωγή των χαρακτηριστικών $F = \mathcal{F}(W)$ του τμηματοποιημένου συνόλου δεδομένων $\{W_i, y_i\}_{i=1}^{N_w}$. Πιο συγκεκριμένα για κάθε τμήμα W_i θα εξαγάγουμε κάποια στατιστικά μεγέθη τα οποία στην συνέχεια θα αποτελέσουν την είσοδο για τον αλγόριθμο της μηχανικής μάθησης. Με αυτήν την ενέργεια από τον χώρο του τμηματοποιημένου σήματος $\{W_i, y_i\}_{i=1}^{N_w}$, μεταφερόμαστε στον χώρο των χαρακτηριστικών $\{F_i, y_i\}_{i=1}^{N_w}$. Αυτή η διαδικασία γίνεται για κάθε τυχαία μεταβλητή του τμήματος W_i ξεχωριστά. Μερικά από τα χαρακτηριστικά (στατιστικά μεγέθη) που χρησιμοποιήθηκαν είναι η μέση τιμή, η διάμεσος, το άθροισμα των τετραγώνων, η τυπική απόκλιση, η διακύμανση, το μέγιστο και ελάχιστο στοιχείο, η λοξότητα, η κύρτωση, η μέση φασματική ενέργεια, το μέσο όρο των απόλυτων τιμών, η ρίζα των μέσων τετραγώνων κα.

5 Προ-επεξεργασία Χαρακτηριστικών

Στην συνέχεια ακολούθησε η κανονικοποίηση των δεδομένων εκπαίδευσης που πλέον είναι τα χαρακτηριστικά που εξαγάγαμε στο προηγούμενο βήμα. Η κανονικοποίηση του συνόλου δεδομένων είναι μια κοινή απαίτηση για πολλούς εκτιμητές μηχανικής μάθησης. Συνήθως αυτό γίνεται αφαιρώντας το μέσο όρο και κλιμακώνοντας τη διακύμανση στη μονάδα. Ωστόσο, οι υπερβολικές τιμές (τα outliers) μπορούν συχνά να επηρεάσουν τη μέση τιμή και την διακύμανση του δείγματος με αρνητικό τρόπο. Για την αντιμετώπιση αυτού το προβλήματος σε αυτό το βήμα διαλέχτηκε η μέθοδος του robust data scaling της βιβλιοθήκης scikit-learn [4] καθώς αγνοεί τις ακραίες τιμές και παράγαγε τα καλύτερα αποτελέσματα σε σχέση με άλλες μεθόδους κανονικοποίησης που δοκιμάστηκαν. Για να το πετύχει αυτό ο συγκεκριμένος μετασχηματισμός λειτουργεί αφαι-

ρώντας την διάμεσο από του χαρακτηριστικού και στην συνέχεια διαιρώντας ενδοτεταρτημοριακό εύρος του (interquartile range).

6 Μείωση της διάστασης

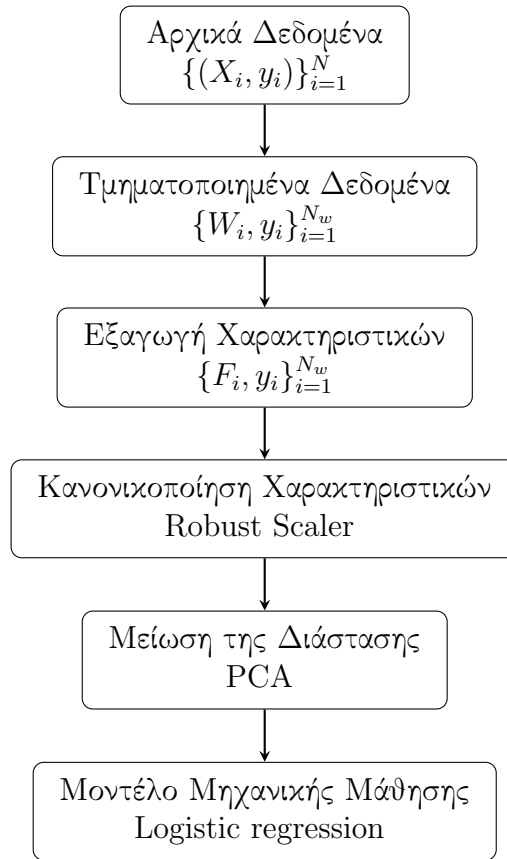
Το αποτέλεσμα της εξαγωγής των χαρακτηριστικών είχε ως αποτέλεσμα την αύξηση της διάστασης των δεδομένων στο χώρο \mathbb{R}^{120} . Αυτό μπορεί να δημιουργήσει διάφορα προβλήματα στους ταξινομητές λόγω του προβλήματος κατάρας της διαστασιμότητας (curse of dimensionality) [5], όπως μείωση της γενικευτική τους ικανότητας καθώς και η ανάγκη για συλλογή περισσότερων δεδομένων. Για να αντιμετωπιστεί αυτό το πρόβλημα, στο συγκεκριμένο βήμα της προ-επεξεργασίας των δεδομένων εφαρμόσαμε την μεθοδολογία PCA (Principal Component Analysis) για την μείωση της διάστασης των δεδομένων [6]. Μέσω της μεθόδου αυτής τα δεδομένα από το χώρο \mathbb{R}^{120} μετασχηματίστηκαν στον χώρο \mathbb{R}^{60} . Η διαδικασία αυτή κατά την αξιολόγηση του τελικού μοντέλου μηχανικής μάθησης φαίνεται ότι βελτίωσε τα αποτελέσματα της μεθόδου.

7 Το μοντέλο

Πλέον μετά από τα στάδια συλλογής και προ-επεξεργασία των δεδομένων, συνεχίζουμε στον ορισμό του μοντέλου μηχανικής μάθησης με στόχο την αυτόματη κατηγοριοποίηση (ταξινόμηση) της πολυδιάστατης χρονοσειράς των αισθητηριακών σημάτων. Δοκιμάστηκαν και εφαρμόστηκαν διαφορετικά μοντέλα μηχανικής μάθησης αλλά αυτό με τα καλύτερα αποτελέσματα φαίνεται να είναι το μοντέλο της λογιστικής παλινδρόμησης (Logistic Regression). Η λογιστική παλινδρόμηση, παρά το γεγονός ότι το όνομά της παραπέμπει στο πρόβλημα της παλινδρόμησης (regression), είναι ένα μοντέλο για ταξινόμηση. Ακολουθεί η μαθηματική έκφραση που ελαχιστοποιεί το λογιστικό μοντέλο μηχανικής μάθησης που χρησιμοποιήθηκε:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1) \quad (1)$$

όπου w άγνωστοι παράμετροι, C υπερπαράμετρος κανονικοποίησης, y_i πραγματική κατηγορία του δεδομένου, X_i τα δεδομένα και c μια σταθερά. Ακολουθεί το σχήμα 2, όπου παραθέτουμε γραφικά όλη την διαδικασία μετασχηματισμού, προ-επεξεργασίας και ταξινόμησης του σήματος, που απαρτίζει το τελικό μοντέλο που κατασκευάστηκε.



Σχήμα 2: Αλυσίδα αυτόματης κατάτμησης και κατηγοριοποίησης της πολυδιάστατης χρονοσειράς αισθητηριακών σημάτων.

8 Βελτιστοποίηση των υπερπαραμέτρων

Υπερπαραμέτροι ονομάζονται οι παράμετροι των μοντέλων μηχανικής μάθησης, οι οποίοι δεν βελτιστοποιούνται κατά την διαδικασία της εκπαίδευσης, αλλά δύνονται σαν είσοδο από το χρήστη. Καθόλη την διάρκεια κατασκευής του μοντέλου, χρειάστηκαν να ορισθούν αρκετές υπερπαραμέτροι που απαιτούσαν οι μεθοδολογίες. Ωστόσο ο τυχαίο ορισμός ή αρχικοποίηση τους δεν είναι βέλτιστος τρόπος να λάβουν τιμή. Για να λύσουμε αυτό το πρόβλημα, κάναμε χρήση της μεθοδολογίας GridSearchCV της βιβλιοθήκης Scikit-learn [4]. Η μεθοδολογία αυτή λαμβάνει για κάθε υπερπαραμέτρο μία λίστα με τιμές. Στην συνέχεια εκπαιδεύει το μοντέλο με κάθε δυνατό συνδυασμό υπερπαραμέτρων και επιστρέφει τον καλύτερο δυνατό συνδυασμό, δηλαδή το μοντέλο με τις υπερπαραμέτρους που παρήγαγαν τα καλύτερα αποτελέσματα στην αξιολόγηση. Στην συνέχεια ακολουθεί ο πίνακας με τις βέλτιστες υπερπαραμέτρους που βρέθηκαν

με την μεθοδολογία GridSearchCV.

Συνάρτηση	Υπερπαράμετρος	Τιμή
Segment	overlap	0.5
Segment	width	50
Robust Scaler	quantile_range	[15.0, 85.0]
PCA	n_components	60
Logistic Regression	C	0.1

Πίνακας 1: Βέλτιστες υπερπαράμετροι του μοντέλου με βάση το GridSearchCV.

9 Αξιολόγηση

Η αξιολόγηση των αποτελεσμάτων της κατηγοριοποίησης (ταξινόμησης) πραγματοποιήθηκε με την χρήση της μεθοδολογίας του cross validation της Scikit-learn [4]. Κατά την μεθοδολογία αυτή, το αρχικό σύνολο δεδομένων διαχωρίζεται σε n ξένα μεταξύ τους υποσύνολα. Στην συνέχεια τα μοντέλα εκπαιδεύονται στα $n - 1$ υποσύνολα (σύνολο εκπαίδευσης) και τελικά ελέγχονται για την ποιότητα των αποτελεσμάτων τους στο τελευταίο υποσύνολο (σύνολο ελέγχου). Η διαδικασία αυτή επαναλαμβάνεται n , έτσι ώστε όλα τα υποσύνολα να χρησιμοποιηθούν για εκπαίδευση αλλά και για έλεγχο. Για την επίτευξη του διαχωρισμού των δεδομένων στο cross validation χρησιμοποιήθηκε η τεχνική group K-Fold που είναι ιδανική όταν το σύνολο δεδομένων αποτελείται από διαφορετικούς χρήστες και εξασφαλίζει ότι ο ίδιος χρήστης ή το ίδιο group χρηστών δεν θα εμφανίζεται ταυτόχρονα σε παραπάνω από ένα υποσύνολα δεδομένων (Folds). Με τον τρόπο αυτό μπορούμε να εκπαιδεύσουμε σε ένα group χρηστών και να μετρήσουμε τις επιδόσεις της μεθόδου σε κάποιο άλλο άγνωστο group. Οι μετρικές που χρησιμοποιήθηκαν για τον έλεγχο της ποιότητας ταξινόμησης είναι το $accuracy \in [0, 1]$ καθώς και το $F1 \in [0, 1]$ score. Τα τελικά $accuracy$ και $F1$ score είναι ο μέσος όρος των n πειραμάτων που πραγματοποιήθηκαν κατά την διαδικασία του group K-Fold cross validation. Ακολουθούν οι σχέσεις των $accuracy$ και $F1$ score :

$$accuracy(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} I(y_i, \hat{y}_i) \quad (2)$$

με n το πλήθος των δεδομένων, y_i η κατηγορία του i -οστού δεδομένου, \hat{y}_i η i -οστή έξοδος του μοντέλου και $I(x, y)$ δείκτρια συνάρτηση όπου ισχύει ότι $I(x, y) := \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$.

$$F1(y, \hat{y}) = 2 * \frac{precision * recall}{precision + recall} \quad (3)$$

$$precision = \frac{truepositive}{truepositive + falsepositive} \quad (4)$$

$$recall = \frac{truepositive}{truepositive + falsenegative} \quad (5)$$

όπου $precision \in [0, 1]$ και $recall \in [0, 1]$. Το $precision$ είναι διαισθητικά η ικανότητα του ταξινομητή να μην χαρακτηρίζει ως θετικό ένα αρνητικό δείγμα ενώ το $recall$ είναι διαισθητικά η ικανότητα του ταξινομητή να βρει όλα τα θετικά δείγματα. Τέλος το $F1$ score είναι ο σταθμισμένος μέσος όρος των $precision$ και $recall$.

10 Πειραματικά αποτελέσματα

Στο παρόν κεφάλαιο παρουσιάζονται τα πειραματικά αποτελέσματα της μεθόδου. Για κάθε χρήστη υπάρχουν επαρκεί δεδομένα για τις δραστηριότητες (κατηγορίες), ...

11 Βιβλιοθήκες που χρησιμοποιήθηκαν

Ο κώδικας γράφηκε σε γλώσσα προγραμματισμού Python 3[7]. Ακολουθούν οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση του προγράμματος.

- Για τους μαθηματικούς υπολογισμούς και τον χειρισμό πινάκων χρησιμοποιήθηκε η βιβλιοθήκη [numpy](#) [8].
- Για τους επιστημονικούς υπολογισμούς χρησιμοποιήθηκε η βιβλιοθήκη [scipy](#) [9].
- Για την επεξεργασία των δεδομένων μέσω `dataframes` χρησιμοποιήθηκε η βιβλιοθήκη [pandas](#) [2].
- Για τα μοντέλα μηχανικής μάθησης έγινε χρήση της βιβλιοθήκης [sklearn](#) [4].
- Η βιβλιοθήκη [matplotlib](#) [10] χρησιμοποιήθηκε για τα γραφήματα.

References

- [1] M. H. C. W. P. H. S. M. David Burns, Nathan Leung, “Shoulder physiotherapy exercise recognition: Machine learning the inertial signals from a smartwatch,” *arXiv*, 2018.
- [2] T. pandas development team, “pandas-dev/pandas: Pandas,” Feb. 2020.
- [3] C. W. David Burns, “Seglearn: A python package for learning sequences and time series,” *arXiv*, 2018.
- [4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [5] C. M. Bishop, “Pattern recognition,” *Machine learning*, vol. 128, no. 9, 2006.
- [6] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [7] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [8] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, Sept. 2020.
- [9] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.

- [10] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.