



Εθνικό Μετσόβιο Πολυτεχνείο (Ε.Μ.Π)  
Σχολή Μηχανολόγων Μηχανικών

ΕΥΦΥΗ ΣΥΣΤΗΜΑΤΑ ΚΑΤΕΡΓΑΣΙΩΝ, ΜΑΘΗΜΑ 9<sup>ΟΤ</sup> ΕΞΑΜΗΝΟΥ

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

---

# Τεχνητά Νευρωνικά Δίκτυα

---

Σπουδαστές:

Λάζαρος Χριστοφορίδης (mc18012)

Μιχάλης Παπαδάκης (mc18026)

Καθηγητής:

Πανώριος Μπερνάρδος

19 Ιουλίου 2025

# Περιεχόμενα

|   |    |
|---|----|
| Περιεχόμενα   | 1  |
| Κατάλογος Σχημάτων  | 2  |
| Κατάλογος Πινάκων   | 2  |
| 1 Εισαγωγή  | 3  |
| 2 Προεπεξεργασία Δεδομένων  | 4  |
| 3 Επιλογή δεδομένων   | 5  |
| 3.1 Επιλογή δεδομένων τραχύτητας . . . . .                            | 5  |
| 3.2 Επιλογή δεδομένων Ορίου διαρροής . . . . .                        | 6  |
| 3.3 Επιλογή δεδομένων παραμόρφωσης . . . . .                          | 7  |
| 4 Βελτιστοποίηση Νευρωνικών δικτύων                                   | 8  |
| 4.1 Διαδικασία Βελτιστοποίησης . . . . .                              | 8  |
| 4.2 Βελτιστοποίηση Νευρωνικού δικτύου πρόβλεψης τραχύτητας . . . . .  | 9  |
| 4.3 Βελτιστοποίηση Νευρωνικού δικτύου πρόβλεψης αντοχής . . . . .     | 11 |
| 4.4 Βελτιστοποίηση Νευρωνικού δικτύου πρόβλεψης επιμήκυνσης . . . . . | 13 |
| 5 Αναφορές  | 16 |
| 6 Κώδικες   | 17 |

## Κατάλογος Σχημάτων

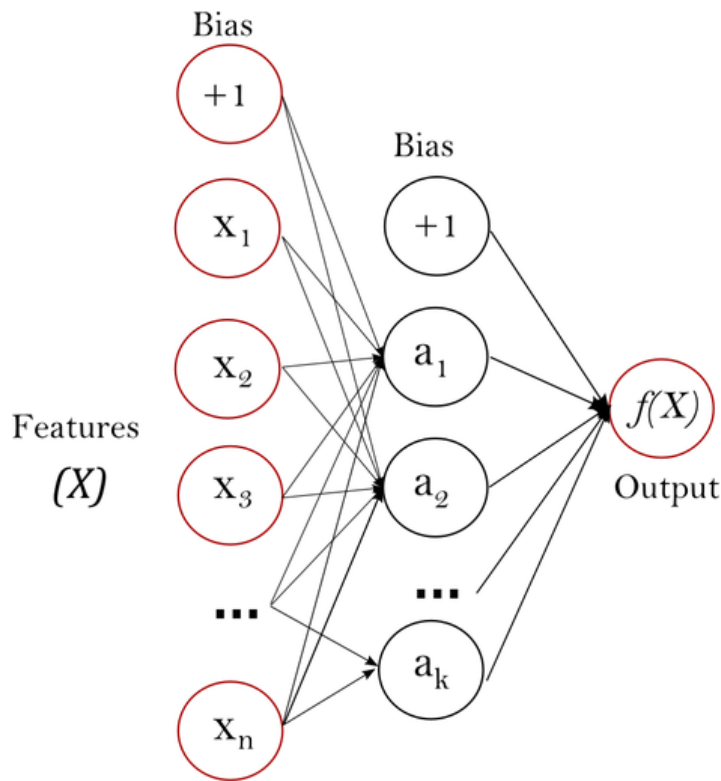
|    |  |    |
|----|--|----|
| 1  | Δομή του <i>Multi-Layer Perceptron</i> που χρησιμοποιείται στην συνάρτηση <i>MLPregressor</i> . . . . .              | 3  |
| 2  | Κωδικοποίηση των κατηγορηματικών τιμών . . . . .   | 4  |
| 3  | Κανονικοποίηση τιμών εισόδων . . . . .   | 4  |
| 4  | Σύγκριση στατιστικών δεικτών $f_{test}$ και $MI$ για την συνάρτηση $y = x_1 + \sin(6\pi x_2) + 0.1N(0, 1)$ . . . . . | 5  |
| 5  | Main effect plot για την μέση τραχύτητα από το άρθρο [3] . . . . .   | 6  |
| 6  | Εξάρτηση τραχύτητας από διάφορες παραμέτρους κατεργασίας σύμφωνα με [7] . . . . .                                    | 6  |
| 7  | Καμπύλες παραμόρφωσης-τάσης για διάφορες τιμές του Infill Density σύμφωνα με [4] . . . . .                           | 7  |
| 8  | Αρχιτεκτονική Νευρωνικού Δικτύου της τραχύτητας . . . . .  | 9  |
| 9  | $R^2$ γράφημα για την τραχύτητα. Με κόκκινο είναι η γραμμή $y = x$ . . . . .   | 10 |
| 10 | Αρχιτεκτονική Νευρωνικού Δικτύου του Ορίου Διαρροής . . . . .  | 11 |
| 11 | $R^2$ γράφημα για το όριο διαρροής. Με κόκκινο είναι η γραμμή $y = x$ . . . . .                                      | 12 |
| 12 | Αρχιτεκτονική Νευρωνικού Δικτύου της παραμόρφωσης . . . . .  | 14 |
| 13 | $R^2$ γράφημα για την παραμόρφωση. Με κόκκινο είναι η γραμμή $y = x$ . . . . .                                       | 15 |

## Κατάλογος Πινάκων

|    |   |    |
|----|---|----|
| 1  | Πίνακας συσχέτισης των παραμέτρων εισόδου . . . . .   | 5  |
| 2  | Πίνακας με τιμές $f_{test}$ , $MI$ για κάθε παράμετρο κατεργασίας σε σχέση με την τραχύτητα . . . . .   | 6  |
| 3  | Πίνακας με τιμές $f_{test}$ , $MI$ για κάθε παράμετρο κατεργασίας σε σχέση με την αντοχή του τεμαχίου . . . . .   | 7  |
| 4  | Πίνακας με τιμές $f_{test}$ , $MI$ για κάθε παράμετρο κατεργασίας σε σχέση με την αντοχή του τεμαχίου . . . . .   | 7  |
| 5  | Απόδοση νευρωνικού δικτύου εκτίμησης τραχύτητας με παραμέτρους εισόδου που αναφέρθηκαν 3 . . . . .  | 9  |
| 6  | Απόδοση νευρωνικού δικτύου εκτίμησης τραχύτητας με προσθήκη στις παραμέτρους εισόδου το πάχος τοιχωμάτων και του υλικού . . . . .                         | 9  |
| 7  | Απόδοση νευρωνικού δικτύου εκτίμησης ορίου διαρροής με παραμέτρους εισόδου που αναφέρθηκαν . . . . .  | 11 |
| 8  | Απόδοση νευρωνικού δικτύου εκτίμησης ορίου διαρροής με προσθήκη στις παραμέτρους το Infill Pattern και το Print speed . . . . .                           | 11 |
| 9  | Απόδοση νευρωνικού δικτύου εκτίμησης επιμήκυνσης με παραμέτρους εισόδου που αναφέρθηκαν . . . . .   | 13 |
| 10 | Απόδοση νευρωνικού δικτύου εκτίμησης επιμήκυνσης με προσθήκη στις παραμέτρους εισόδου το Wall Thickness το Infill Density και το Infill Pattern . . . . . | 13 |
| 11 | Απόδοση νευρωνικού δικτύου εκτίμησης επιμήκυνσης με προσθήκη στις παραμέτρους εισόδου το Wall Thickness και το Infill Density . . . . .                   | 13 |

## 1. Εισαγωγή

Σκοπός της εργασίας ήταν η βελτιστοποίηση της αρχιτεκτονικής ενός νευρωνικού δικτύου για την πρόβλεψη της τραχύτητας και των μηχανικών ιδιοτήτων (Αντοχή και παραμόρφωση) τεμαχίων που παράχθηκαν με *3D printing*. Συνολικά δημιουργήθηκαν 3 διαφορετικά νευρωνικά δίκτυα για καθένα χαρακτηριστικό. Παράγοντες διερεύνησης αποτελούν η επιλογή των παραμέτρων εισόδου, ο αριθμός των κρυμμένων επιπέδων και των κρυμμένων νευρώνων σε αυτά και τέλος το κριτήριο απόδοσης. Το νευρωνικό δίκτυο υλοποιήθηκε στην python με χρήση του πακέτου *scikit-learn*<sup>1</sup>. Χρησιμοποιήθηκε το μοντέλο *MLPRegressor*<sup>2</sup>, το οποίο είναι ένα νευρωνικό πρόσθιας τροφοδότησης - οπισθοδρόμης διάδοσης με δομή που παρουσιάζεται στην εικόνα 1.



Σχήμα 1: Δομή του *Multi-Layer Perceptron* που χρησιμοποιείται στην συνάρτηση *MLPRegressor*.

Το πρώτο βήμα ήταν η προ-επεξεργασία των δεδομένων. Έπειτα με διάφορους στατιστικούς δείκτες και από την βιβλιογραφία διερευνήθηκε ποιοι παράμετροι πρέπει να εισαχθούν σε καθένα από τα δίκτυα. Καθώς όμως η απόδοση του δικτύου εξαρτάται τόσο από τα δεδομένα εισόδου, από την αρχιτεκτονική του αλλά και από την αρχικοποίηση, δημιουργήθηκε ένα πρόγραμμα που ελέγχει διάφορους συνδυασμούς παραμέτρων με διαφορετικές αρχιτεκτονικές για πολλαπλές αρχικοποιήσεις. Λόγω της έμφυτης τυχαιότητας, το βέλτιστο δίκτυο προκύπτει χρησιμοποιώντας το Welch's t-test, ώστε τα συμπεράσματά ως προς την διαφορετικά απόδοση αρχιτεκτονικών να έχουν στατιστική σημαντικότητα.

<sup>1</sup><https://scikit-learn.org/stable/>

<sup>2</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPRegressor.html#sklearn.neural\\_network.MLPRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html#sklearn.neural_network.MLPRegressor)

## 2. Προεπεξεργασία Δεδομένων

Το πρώτο βήμα ήταν η προ-επεξεργασία των δεδομένων. Αρχικά, οι κατηγορηματικές τιμές (Infill pattern, material) κωδικοποιήθηκαν σε αριθμητικές και αποθηκεύτηκαν σε ένα νέο αρχείο *data.csv*. Καθώς υπήρχαν μόνο 2 κατηγορίες σε κάθε feature, οι κατηγορίες αντιστοιχίστηκαν σε 0 και 1<sup>3</sup>.

| 1 | layer_heig | wall_thicki | infill_densi | infill_patten | nozzle_ter | bed_tempi | print_spee | material | fan_speed | roughness | tension_st | elongation (%) |
|---|------------|-------------|--------------|---------------|------------|-----------|------------|----------|-----------|-----------|------------|----------------|
| 2 | 0.02       | 8           | 90           | 1             | 220        | 60        | 40         | 1        | 0         | 25        | 18         | 1.2            |
| 3 | 0.02       | 7           | 90           | 0             | 225        | 65        | 40         | 1        | 25        | 32        | 16         | 1.4            |
| 4 | 0.02       | 1           | 80           | 1             | 230        | 70        | 40         | 1        | 50        | 40        | 8          | 0.8            |
| 5 | 0.02       | 4           | 70           | 0             | 240        | 75        | 40         | 1        | 75        | 68        | 10         | 0.5            |
| 6 | 0.02       | 6           | 90           | 1             | 250        | 80        | 40         | 1        | 100       | 92        | 5          | 0.7            |
| 7 | 0.02       | 10          | 40           | 0             | 200        | 60        | 40         | 0        | 0         | 60        | 24         | 1.1            |

Σχήμα 2: Κωδικοποίηση των κατηγορηματικών τιμών

Ύστερα, έγινε κανονικοποίηση (standardization) των υπόλοιπων μεταβλητών, καθώς είχαν διαφορετική τάξη μεγέθους. Για την κανονικοποίηση χρησιμοποιήθηκε η συνάρτηση *standard scaler*<sup>4</sup>.

| layer_heig | wall_thicki | infill_densi | infill_patté | nozzle_ter | bed_tempi | print_spee | material | fan_speed | roughness | tension_st | elongation (%) |
|------------|-------------|--------------|--------------|------------|-----------|------------|----------|-----------|-----------|------------|----------------|
| -1.34903   | 0.960817    | 1.45767      | 1            | -0.10224   | -1.41421  | -0.8165    | 1        | -1.41421  | 25        | 18         | 1.2            |
| -1.34903   | 0.615199    | 1.45767      | -1           | 0.238559   | -0.70711  | -0.8165    | 1        | -0.70711  | 32        | 16         | 1.4            |
| -1.34903   | -1.45851    | 1.059399     | 1            | 0.579358   | 0         | -0.8165    | 1        | 0         | 40        | 8          | 0.8            |
| -1.34903   | -0.42165    | 0.661129     | -1           | 1.260957   | 0.707107  | -0.8165    | 1        | 0.707107  | 68        | 10         | 0.5            |
| -1.34903   | 0.269582    | 1.45767      | 1            | 1.942555   | 1.414214  | -0.8165    | 1        | 1.414214  | 92        | 5          | 0.7            |
| -1.34903   | 1.652052    | -0.53368     | -1           | -1.46544   | -1.41421  | -0.8165    | -1       | -1.41421  | 60        | 24         | 1.1            |

Σχήμα 3: Κανονικοποίηση τιμών εισόδων

Ύστερα, πρέπει να γίνει χωρισμός των δεδομένων σε train, validation και test υποσύνολα. Επειδή το dataset ήταν αρκετά μικρό, δεν υπήρξε υποσύνολο επικύρωσης. Ο χωρισμός έγινε με την συνάρτηση *train\_test\_split*<sup>5</sup>, και το μέγεθος του υποσυνόλου. Για να μην βρεθεί κάποιο νευρωνικό δίκτυο το οποίο προσαρμόζεται σε συγκεκριμένα δεδομένα, κάθε φορά γίνεται διαφορετικός χωρισμός.

<sup>3</sup>Σε αυτή την περίπτωση το *one hot encoding* είναι ισοδύναμο με το *label encoding*.

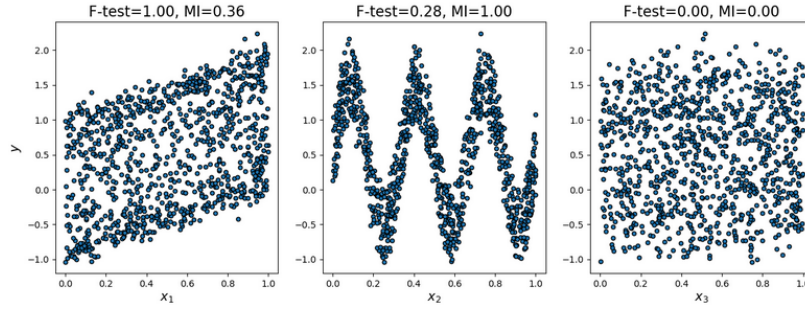
<sup>4</sup><https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

<sup>5</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.train\\_test\\_split.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)

### 3. Επιλογή δεδομένων

Το dataset αποτελείται από 9 παραμέτρους κατεργασίας, ωστόσο δεν είναι απαραίτητο ότι όλες οι παράμετροι επηρεάζουν όλα τα χαρακτηριστικά. Η μεθοδολογία που ακολουθήθηκε για την επιλογή των παραμέτρων εισόδου στο νευρωνικό δίκτυο είναι η εξής:

- Χρήση στατιστικών εργαλείων ώστε να εξετασθεί η συσχέτιση κάποιας παραμέτρου με κάθε χαρακτηριστικό. Συγκεκριμένα χρησιμοποιήθηκαν οι στατιστικοί δείκτες  $f_{\text{regression}}$  ( $f_{\text{test}}$ ) και  $\text{mutual information}(MI)$ . Ο πρώτος δείκτης είναι παρόμοιος της γραμμικής συσχέτισης ( $\rho_{x,y}$ ) και εξετάζει αν μια παράμετρος έχει γραμμική συσχέτιση με κάποιο χαρακτηριστικό. Ο δεύτερος δείκτης είναι ένα μέτρο της γενικότερης εξάρτησης των δυο μεταβλητών. Όπως φαίνεται από την εικόνα 4, ο δείκτης  $f_{\text{test}}$  πιάνει έντονα την γραμμική συσχέτιση ενώ ο δείκτης  $MI$  οποιαδήποτε σχέση<sup>6</sup>.



Σχήμα 4: Σύγκριση στατιστικών δεικτών  $f_{\text{test}}$  και  $MI$  για την συνάρτηση  $y = x_1 + \sin(6\pi x_2) + 0.1N(0,1)$

- Εύρεση βιβλιογραφίας που δείχνει εξάρτηση ενός χαρακτηριστικού από παραμέτρους κατεργασίας.
- Διερεύνηση της συσχέτισης των δεδομένων εισόδου. Ο πίνακας συσχέτισης φαίνεται στον πίνακα 1. Παρατηρείται πως η ταχύτητα του ανεμιστήρα ψύξης και η θερμοκρασία του κρεβατιού παρουσιάζουν έντονη συσχέτιση, και μεταβάλλονται με τον ίδιο ρυθμό ( $\rho_{\text{bedtemperature}, \text{fanspeed}} = 1$ ). Έτσι, επιλέχθηκε η θερμοκρασία του κρεβατιού να αφαιρεθεί από τις παραμέτρους εισόδου, καθώς δεν θα πρόσθετε παραπάνω πληροφορία στο νευρωνικό δίκτυο. **Όποιο δίκτυο εκπαιδευτεί με την ταχύτητα του ανεμιστήρα θα μπορούσε να εκπαιδευτεί με την θερμοκρασία κρεβατιού.**

| $\rho_{x,y}$ | 1         | 2         | 3         | 5         | 6         | 7         | 9         |
|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1            | 1         | -1.93E-01 | 3.50E-03  | 9.79E-18  | -7.39E-18 | -5.55E-02 | 6.40E-18  |
| 2            | -1.93E-01 | 1         | 1.03E-01  | -1.18E-01 | -2.93E-02 | -4.20E-01 | -2.93E-02 |
| 3            | 3.50E-03  | 1.03E-01  | 1         | 2.39E-01  | 1.60E-18  | -9.43E-02 | 1.60E-18  |
| 5            | 9.79E-18  | -1.18E-01 | 2.39E-01  | 1         | 6.02E-01  | 0         | 6.02E-01  |
| 6            | -7.39E-18 | -2.93E-02 | 1.60E-18  | 6.02E-01  | 1         | 0.00E+00  | 1         |
| 7            | -5.55E-02 | -4.20E-01 | -9.43E-02 | 0         | 0.00E+00  | 1         | 0.00E+00  |
| 9            | 6.40E-18  | -2.93E-02 | 1.60E-18  | 6.02E-01  | 1         | 0.00E+00  | 1         |

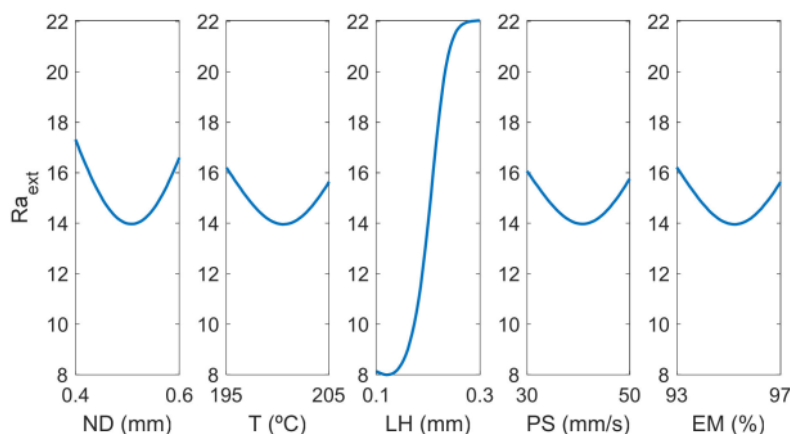
Πίνακας 1: Πίνακας συσχέτισης των παραμέτρων εισόδου

#### 3.1 Επιλογή δεδομένων τραχύτητας

Αρχικά από την βιβλιογραφία, παρατηρήθηκε πως οι παράμετροι οι οποίοι επηρεάζουν την τραχύτητα είναι το *layer height*, η κλίση του τεμαχίου και η διάμετρος του nozzle [1], [2], [3]. Αυτό φαίνεται και στην εικόνα 5 από το [3]. Ωστόσο, στις παραμέτρους του datasheet δεν υπήρχε η διάμετρος του ακροφυσίου και η κλίση του τεμαχίου ούτε κάποιο άλλο παρεμφερές χαρακτηριστικό προς αυτά.

Έστερα, υπολογίστηκαν οι στατιστικοί δείκτες  $f_{\text{test}}$ ,  $MI$  για κάθε παράμετρο κατεργασίας και παρουσιάζονται στον πίνακα 2. Παρατηρείται πως η τραχύτητα έχει έντονη εξάρτηση από το *layer height* αλλά, και την θερμοκρασία του ακροφυσίου και της ταχύτητας εκτύπωσης. Διαισθητικά, η εξάρτηση από την θερμοκρασία του ακροφυσίου είναι λογική, αφού επηρεάζει τα

<sup>6</sup>[https://scikit-learn.org/stable/auto\\_examples/feature\\_selection/plot\\_f\\_test\\_vs\\_mi.html#sphx-glr-auto-examples-feature-selection-plot-f-test-vs-mi-py](https://scikit-learn.org/stable/auto_examples/feature_selection/plot_f_test_vs_mi.html#sphx-glr-auto-examples-feature-selection-plot-f-test-vs-mi-py)



Σχήμα 5: Main effect plot για την μέση τραχύτητα από το άρθρο [3]

θερμοκρασιακά πεδία κατά την στερεοποίηση του πλαστικού και την ένωση με τα υπόλοιπα layers ενώ η ταχύτητα εκτύπωσης επηρεάζει και την θερμοκρασία, αλλά αυξημένη ταχύτητα οδηγεί και σε περισσότερες ταλαντώσεις στο σύστημα κίνησης που ενδέχεται να αυξάνουν την τραχύτητα. Παρατηρούνται σχετικά αυξημένες τιμές και στο πάχος των τοιχωμάτων και το υλικό. Αυτοί οι παράμετροι αρχικά αγνοήθηκαν αλλά στη συνέχεια ελέγχθηκε αν βελτιώνουν την απόδοση του δικτύου.

| Feature:     | 1             | 2            | 3     | 4     | 5            | 6     | 7            | 8            | 9     |
|--------------|---------------|--------------|-------|-------|--------------|-------|--------------|--------------|-------|
| <b>Ftest</b> | <b>86.133</b> | <u>2.607</u> | 0.682 | 0.225 | <b>6.640</b> | 1.840 | 0.714        | <u>2.760</u> | 1.840 |
| <b>MI</b>    | <b>0.744</b>  | 0.002        | 0     | 0     | 0            | 0     | <b>0.339</b> | 0            | 0     |

Πίνακας 2: Πίνακας με τιμές *f*test, *MI* για κάθε παράμετρο κατεργασίας σε σχέση με την τραχύτητα

### 3.2 Επιλογή δεδομένων Ορίου διαρροής

Στον παρακάτω πίνακα φαίνονται, οι συσχετίσεις των παραμέτρων εκτύπωσης με το όριο διαρροής ( $\sigma_{max}$ ), από πειράματα που έγιναν στο πανεπιστήμιο της Καταλωνίας (Ισπανία) [7]

Table 6. Significance value of the parameters with respect to the answers.

| Factor            | Elastic Properties  |                        | Plastic Properties                  |                                    |
|-------------------|---------------------|------------------------|-------------------------------------|------------------------------------|
|                   | Young's Modulus (E) | Yield Strength (Rp0.2) | Maximum Strength ( $\sigma_{max}$ ) | Maximum Deformation ( $\epsilon$ ) |
| Layer orientation | ✓✓✓                 | ✓✓✓                    | ✓✓✓                                 | ✓✓✓                                |
| Layer Height      | ✓✓                  | ✓✓                     | ✓✓                                  | ✗                                  |
| Filament width    | ✓✓                  | ✓                      | ✓✓                                  | ✗                                  |
| Printing velocity | ✓                   | ✓                      | ✓                                   | ✗                                  |
| Infill density    | ✗                   | ✗                      | ✗                                   | ✗                                  |
| Infill pattern    | ✗                   | ✗                      | ✗                                   | ✗                                  |

Σχήμα 6: Εξάρτηση τραχύτητας από διάφορες παραμέτρους κατεργασίας σύμφωνα με [7]

Βάση της παραπάνω δημοσίευσης, σημαντικότερη παράμετρος στην αντοχή του τελικού προϊόν είναι ο προσανατολισμός εκτύπωσης. Έπειτα ακολουθούν οι παράμετροι Layer Height και Filament width (Wall thickness) και σημαντικότερη συσχέτιση έχει το Printing Speed. Τέλος η συγκεκριμένη δημοσίευση καταλήγει στο συμπέρασμα ότι το Infill Density και Infill Pattern δεν έχουν συσχέτιση στο όριο διαρροής. Μια άλλη δημοσίευση [6] καταλήγει στο προφανές συμπέρασμα ότι το Infill Density και το Material έχουν επιρροή στη αντοχή του τεμαχίου. Τέλος μια ακόμα δημοσίευση [5] επιβεβαιώνει το γεγονός πως το Wall Thickness και το External (Nozzle) Temperature έχουν σημαντική συσχέτιση με το όριο διαρροής, το παρακάτω πόρισμα επιβεβαιώνεται και από τα στατιστικά τεστ που γίνονται παρακάτω.

Υστερα, υπολογίστηκαν οι στατιστικοί δείκτες *f*test, *MI* για κάθε παράμετρο κατεργασίας και παρουσιάζονται στον πίνακα 3. Παρατηρείται πως η αντοχή του τεμαχίου δεν εξαρτάται από το Infill Pattern καθώς και οι δύο δείκτες είναι αρκετά μικροί.

Επιπλέον, αρχικά αγνοήθηκε το printing speed καθώς και οι δύο δείκτες  $f_{test}$ ,  $MI$  είναι σχετικά μικροί. Όμως, θα ελεγχθεί αν βελτιώνει την απόδοση, διότι σύμφωνα με την δημοσίευση [7] επιδρά στο όριο διαρροής.

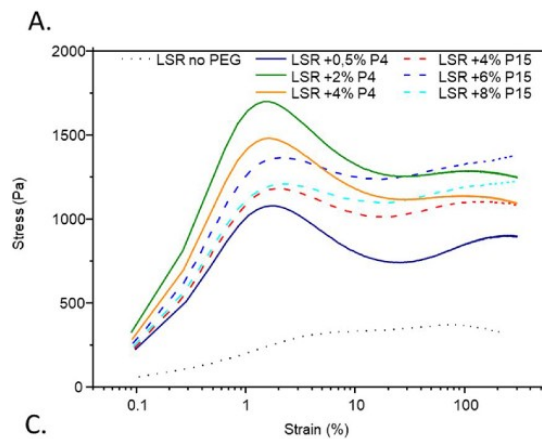
| Feature:     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>Ftest</b> | 6.2   | 9.135 | 7.077 | 0.004 | 9.469 | 3.279 | 3.613 | 4.398 | 3.279 |
| <b>MI</b>    | 0.106 | 0.142 | 0.344 | 0     | 0.091 | 0.063 | 0.011 | 0.097 | 0.054 |

Πίνακας 3: Πίνακας με τιμές  $f_{test}$ ,  $MI$  για κάθε παράμετρο κατεργασίας σε σχέση με την αντοχή του τεμαχίου

### 3.3 Επιλογή δεδομένων παραμόρφωσης

Από το σχήμα 6 παρατηρείται πως η παραμόρφωση εξαρτάται κυρίως από την διεύθυνση με την οποία τυπώνεται το εξάρτημα. Αυτό είναι αναμενόμενο, διότι η παραμόρφωση του τεμαχίου οφείλεται κυρίως στην παραμόρφωση των ινών του. Ωστόσο, δεν υπάρχει σχετική παράμετρος στο dataset.

Ως γνωστόν, από την μηχανική, κάθε υλικό περιγράφεται από την χαρακτηριστική της καμπύλη παραμόρφωσης-τάσης  $\epsilon-\sigma$ . Στο παρακάτω σχήμα φαίνονται διάφορες χαρακτηριστικές υλικών για διάφορα ποσοστά Infill Density, και παρατηρείται πως η μέγιστη παραμόρφωση είναι ανεξάρτητη του παράγοντα αυτού [4].



Σχήμα 7: Καμπύλες παραμόρφωσης-τάσης για διάφορες τιμές του Infill Density σύμφωνα με [4]

Στον πίνακα 4 παρουσιάζονται τα στατιστικά τεστ συσχέτισης των εισόδων με την παραμόρφωση. Αμέσως παρατηρείται πως το Wall Thickness, το Infill Pattern και το Infill Density έχουν μηδενικές τιμές  $MI$  και χαμηλές τιμές  $F_{test}$ . Μόνο το Layer Height και το Nozzle temperature έχουν υψηλές τιμές  $F_{test}$ . Επιπλέον, είναι γνωστό από τη μηχανική πως το η παραμόρφωση εξαρτάται από το υλικό. Μάλιστα, η παραμόρφωση στα 3D prints οφείλεται στην παραμόρφωση των ινών. Οπότε, το υλικό είναι μια παράμετρος που θα εισαχθεί στα δεδομένα εισόδου. Τέλος, η ταχύτητα του ανεμιστήρα, επίσης επιλέχθηκε να εισαχθεί στα δεδομένα εισόδου.

| Feature:     | 1      | 2            | 3            | 4            | 5      | 6     | 7     | 8     | 9     |
|--------------|--------|--------------|--------------|--------------|--------|-------|-------|-------|-------|
| <b>Ftest</b> | 16.659 | <b>1.541</b> | <b>1.245</b> | <b>0.102</b> | 18.500 | 4.778 | 2.782 | 8.860 | 4.778 |
| <b>MI</b>    | 0.180  | 0.060        | <b>0.000</b> | <b>0.003</b> | 0.201  | 0.101 | 0.077 | 0.074 | 0.105 |

Πίνακας 4: Πίνακας με τιμές  $f_{test}$ ,  $MI$  για κάθε παράμετρο κατεργασίας σε σχέση με την αντοχή του τεμαχίου



## 4. Βελτιστοποίηση Νευρωνικών δικτύων

### 4.1 Διαδικασία Βελτιστοποίησης

Οι υποψήφιες αρχιτεκτονικές που εξετάζονται είναι οι ακόλουθες

- Ένα hidden layer με νευρώνες 1 έως 51 (50 συνδυασμοί)
- Δύο hidden layers με νευρώνες σε κάθε layer 1 έως 31 με βήμα 3 (10x10 συνδυασμοί)

Καθώς το dataset αποτελείται από 50 δείγματα, είναι επιθυμητό να μην ξεπεραστούν τα 100 βάρη, ώστε να μην γίνει overfitting και για να συγκλίνει καλύτερα το δίκτυο. Για να αξιολογηθεί το Ν.Δ. ανεξάρτητα των τυχαίων training samples, εκπαιδεύονται  $k$  Ν.Δ. ίδιας αρχιτεκτονικής με διαφορετικά κάθε φορά training samples (RandomFactor=i).

Η αξιολόγηση του κάθε νευρωνικού δικτύου, έγινε βάση του συντελεστή προσδιορισμού  $R^2$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_{target,i} - y_{predicted,i})^2}{\sum_{i=1}^n (y_{target,i} - \bar{y})^2} \quad (1)$$

Υστερα, υπολογίζεται η μέση τιμή ( $\mu$ ) και η τυπική απόκλιση ( $\sigma$ ) των παραπάνω  $k$  δειγμάτων:

$$\mu = \frac{\sum_{i=1}^k R_i^2}{k}, \quad \sigma = \sqrt{\frac{\sum_{i=1}^k (R_i^2 - \bar{R}^2)^2}{k - 1}} \quad (2)$$

Το καλύτερο νευρωνικό δίκτυο είναι αυτό με την μεγαλύτερη μέση τιμή του συντελεστή  $R^2$  ( $R^2 \in (-\infty, 1]$ ). Όμως οι μέσες τιμές δεν μπορούν να συγκριθούν απευθείας, λόγω της έμφυτης τυχαιότητας στην διαδικασία της εκπαίδευσης. Επειδή κάθε συνδυασμός-αρχιτεκτονική αναμένεται να έχει διαφορετική μέση τιμή και διαφορετική τυπική απόκλιση, επιλέχθηκε να χρησιμοποιηθεί το Welch's t-test<sup>7</sup> για να συγκριθεί αν ένας συνδυασμός-αρχιτεκτονική έχει καλύτερη απόδοση με στατιστική σημαντικότητα. Στην πράξη, δημιουργείται ένας συγκεντρωτικός πίνακας με τις μέσες τιμές και την τυπική απόκλιση κάθε αρχιτεκτονικής και ξεκινώντας από τον πρώτο συνδυασμό, ελέγχεται αν κάθε επόμενος είναι καλύτερος με στατιστική σημαντικότητα. Αν είναι, τότε αποθηκεύεται αυτός ως καλύτερος και οι επόμενες συγκρίσεις γίνονται με αυτόν. Το επίπεδο της σημαντικότητας τέθηκε αυθαίρετα στο 0.15%.<sup>8</sup> Συνοπτικά, η διαδικασία της σύγκρισης παρουσιάζεται παρακάτω (Δείκτης τρέχοντα καλύτερου συνδυασμού: *best, now*, Δείκτης συνδυασμού υπό διερεύνηση: *i*):

1.  $H_0 = \mu_i - \mu_{Best,now} = 0$
2.  $H_1 = \mu_i - \mu_{Best,now} > 0$
3. Εύρεση του  $p - value$  από το Welch's t test. Χρήση της συνάρτησης `scipy.stats.ttest_ind_from_stats`<sup>9</sup>
4. Σύγκριση  $p - value < 0.15$ . Αν είναι μικρότερο, τότε απορρίπτεται η μηδενική υπόθεση και ο συνδυασμός  $i$  έχει καλύτερη απόδοση με στατιστική σημαντικότητα 15%.
5. Αλλαγή δεικτών:  $best, now = i$

<sup>7</sup>[https://en.wikipedia.org/wiki/Welch%27s\\_t-test](https://en.wikipedia.org/wiki/Welch%27s_t-test)

<sup>8</sup>Τα δεδομένα είναι λίγα για να απαιτηθεί ένα αυστηρό επίπεδο σημαντικότητας όπως το 0.05.

<sup>9</sup>Documentation βρίσκεται στη σελίδα: [https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest\\_ind\\_from\\_stats.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind_from_stats.html). Θέτοντας ως παράμετρο `equal_var = False`, και `alternative='greater'` επιλέγεται να γίνει το Welch's t-test που προαναφέρθηκε.

## 4.2 Βελτιστοποίηση Νευρωνικού δικτύου πρόβλεψης τραχύτητας

Αρχικά εκπαιδεύτηκε το Ν.Δ. με παραμέτρους εισόδους, που επιλέχθηκαν από τα στατιστικά τεστ συσχέτισης

$$arr = [1, 5, 7] = [layer\_height, nozzle\_temperature, print\_speed] \quad (3)$$

Στον παρακάτω πίνακα φαίνονται οι καλύτερες αρχιτεκτονικές κατά μέση τιμή, καθώς και η στατιστικά ισοδύναμη καλύτερη λύση.

| Solutions                          | Layer1 | Layer2 | $\mu$ | $\sigma$ |
|------------------------------------|--------|--------|-------|----------|
| Κατάταξη κατά μέση τιμή            |        |        |       |          |
| #1                                 | 19     | 22     | 0.908 | 0.132    |
| #2                                 | 28     | 28     | 0.902 | 0.131    |
| #3                                 | 28     | 13     | 0.895 | 0.130    |
| #4                                 | 19     | 28     | 0.895 | 0.129    |
| #5                                 | 19     | 25     | 0.894 | 0.129    |
| Στατιστικά ισοδύναμη καλύτερη λύση |        |        |       |          |
| #1                                 | 7      | 10     | 0.845 | 0.118    |

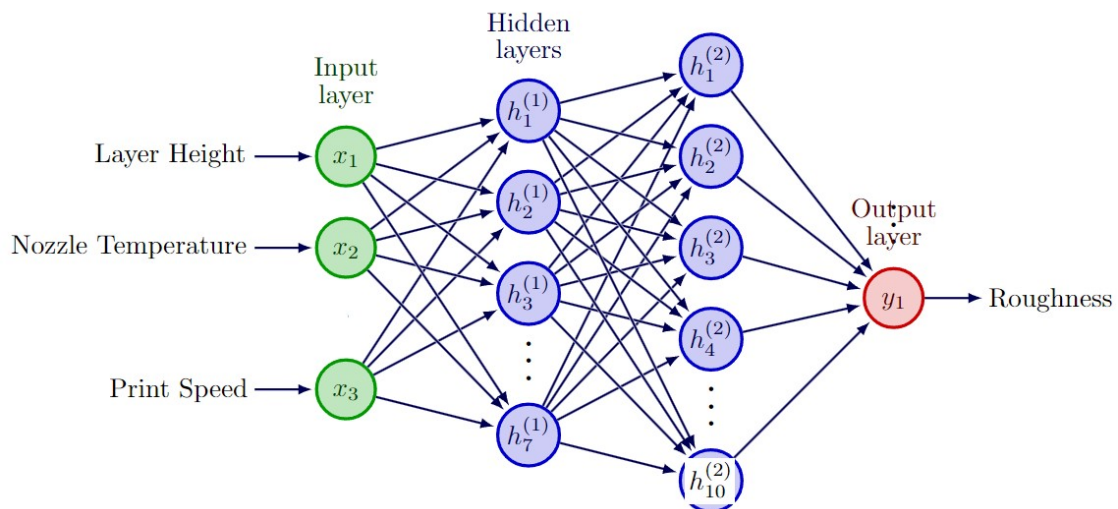
Πίνακας 5: Απόδοση νευρωνικού δικτύου εκτίμησης τραχύτητας με παραμέτρους εισόδου που αναφέρθηκαν 3

Έπειτα, έγινε πρόσθεση των παρακάτω παραμέτρων στις εισόδους, ώστε να γίνει έλεγχος αν γίνεται βελτίωση στην απόδοση του Ν.Δ.

| Solutions   | Layer1 | Layer2 | $\mu$ | $\sigma$ |
|---|--------|--------|-------|----------|
| Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη wall thickness |        |        |       |          |
| #1  | 4      | 7      | 0.500 | 0.064    |
| Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη material       |        |        |       |          |
| #1  | 17     | 0      | 0.644 | 0.082    |

Πίνακας 6: Απόδοση νευρωνικού δικτύου εκτίμησης τραχύτητας με προσθήκη στις παραμέτρους εισόδου το πάχος τοιχωμάτων και του υλικού

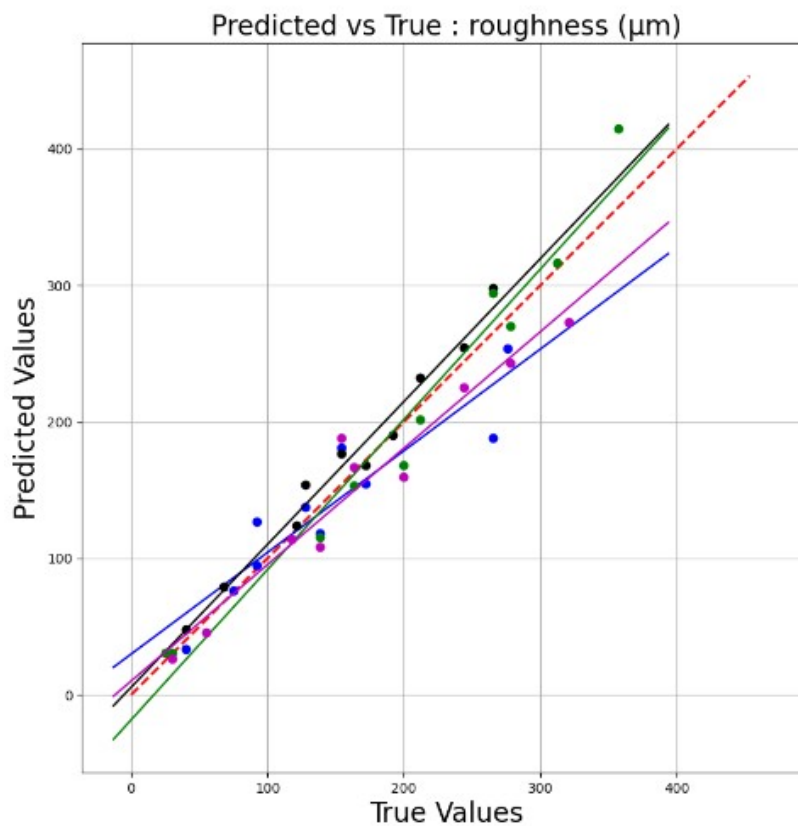
Συγκρίνοντας τους πίνακες 5 και 6 παρατηρείται πως η προσθήκη του πάχους τοιχώματος και του υλικού δεν βελτιώνει την εκτίμηση του νευρωνικού δικτύου. Επιπλέον, ως βέλτιστη αρχιτεκτονική επιλέγεται ένα δίκτυο με 2 layers τα οποία έχουν 7 και 10 κρυφούς νευρώνες το καθένα (Αρχιτεκτονική (7, 10) ).



Σχήμα 8: Αρχιτεκτονική Νευρωνικού Δικτύου της τραχύτητας

Αυτή η αρχιτεκτονική αντιστοιχεί σε 101 βάρη ( $3x7 + 7x10 + 10x1 = 101$ ). Όπως φαίνεται στον πίνακα 5, υπήρξαν δίκτυα με  $R^2 \approx 0.9$  (πχ. Αρχιτεκτονική (19, 22) ). Ωστόσο, αυτή η αρχιτεκτονική δεν είναι απαραίτητα καλύτερη από την (7, 10), διότι δεν υπάρχει ο απαραίτητος αριθμός δειγμάτων για να προκύψει αυτό το συμπέρασμα με στατιστική σημαντικότητα. Επιπλέον, ο μεγάλος αριθμός βαρών ( $3x19 + 19x22 + 22 = 497$ ) σε σχέση με τα δεδομένα αυξάνει σημαντικά την πιθανότητα overfitting.

Στο σχήμα 9 παρουσιάζεται ένα  $R^2$  plot, στο οποίο συγκρίνονται οι πραγματικές (True) με τις εκτιμώμενες από το νευρωνικό δίκτυο (Predicted) τιμές τραχύτητας για παραμέτρους εισόδου από το υποσύνολο αξιολόγησης (test dataset). Για την γενίκευση των αποτελεσμάτων, παρουσιάζεται το γράφημα αυτό για τέσσερα νευρωνικά δίκτυα ίδιας αρχιτεκτονικής ( Αρχιτεκτονική (7,10) ) τα οποία έχουν εκπαιδευτεί σε διαφορετικό υποσύνολο παραμέτρων train dataset και συνεπώς αξιολογούνται στο συμπληρωματικό (test dataset). Η κόκκινη διακεκομμένη γραμμή αντιστοιχεί στην γραμμή  $y = x$ . Ένα τέλειο νευρωνικό δίκτυο που θα εκτιμούσε τις πραγματικές τιμές, θα είχε ένα  $R^2$  plot με σημεία πάνω σε αυτή την ευθεία. Οι υπόλοιπες ευθείες έχουν προκύψει από γραμμική παλινδρόμηση των εκτιμώμενων και πραγματικών τιμών κάθε σετ αξιολόγησης. Παρατηρείται πως σε όλο το εύρος των τιμών τραχύτητας, οι εκτιμήσεις είναι κοντά στις πραγματικές τιμές. Στις πολύ μεγάλες τιμές υπάρχει μεγαλύτερη απόκλιση, η οποία είναι αναμενόμενη καθώς υπήρχαν και λιγότερα δεδομένα με μεγάλη τραχύτητα<sup>10</sup>.



Σχήμα 9:  $R^2$  γράφημα για την τραχύτητα. Με κόκκινο είναι η γραμμή  $y = x$

<sup>10</sup>Υπήρχαν συνολικά 6 tests με ( $Ra > 300\mu m$ ) ενώ υπήρχαν 15 tests με ( $Ra < 100\mu m$ )

### 4.3 Βελτιστοποίηση Νευρωνικού δικτύου πρόβλεψης αντοχής

Αρχικά εκπαιδεύτηκε το Ν.Δ. με παραμέτρους εισόδου, που επιλέχθηκαν από τα στατιστικά τεστ συσχέτισης

$$arr = [1, 2, 3, 5, 8, 9] = [layer\_height, wall\_thickness, infill\_density, nozzle\_temperature, material, fan\_speed] \quad (4)$$

Στον Πίνακα 7 φαίνονται οι καλύτερες αρχιτεκτονικές κατά μέση τιμή, καθώς και η στατιστικά ισοδύναμη καλύτερη λύση.

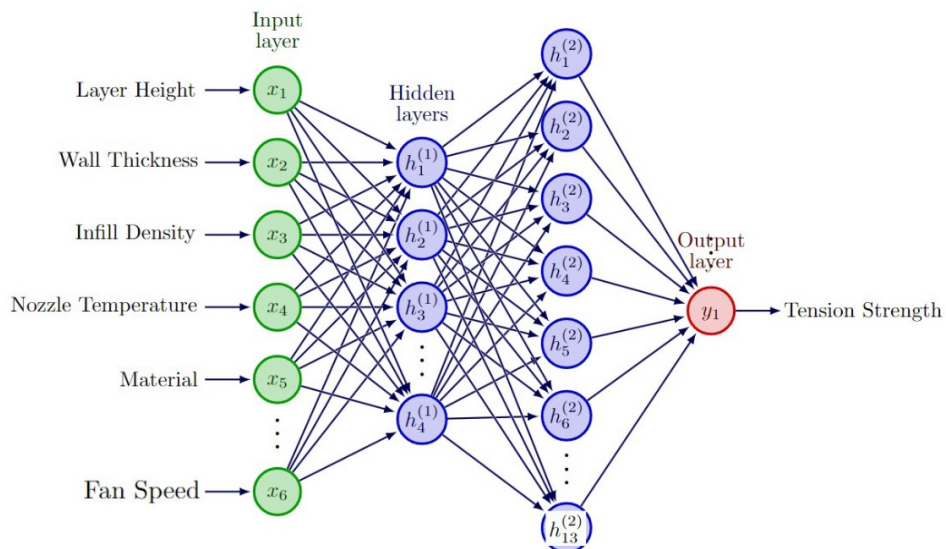
| Solutions                          | Layer1 | Layer2 | $\mu$ | $\sigma$ |
|------------------------------------|--------|--------|-------|----------|
| Κατάταξη κατά μέση τιμή            |        |        |       |          |
| #1                                 | 4      | 13     | 0.743 | 0.00897  |
| #2                                 | 4      | 7      | 0.698 | 0.0706   |
| #3                                 | 9      | 0      | 0.649 | 0.00473  |
| #4                                 | 4      | 16     | 0.647 | 0.0526   |
| #5                                 | 13     | 7      | 0.606 | 0.0412   |
| Στατιστικά ισοδύναμη καλύτερη λύση |        |        |       |          |
| #1                                 | 4      | 13     | 0.743 | 0.00897  |

Πίνακας 7: Απόδοση νευρωνικού δικτύου εκτίμησης ορίου διαρροής με παραμέτρους εισόδου που αναφέρθηκαν

| Solutions   | Layer1 | Layer2 | $\mu$  | $\sigma$ |
|---|--------|--------|--------|----------|
| Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη Infill Pattern |        |        |        |          |
| #1  | 22     | 1      | 0.621  | 0.0162   |
| Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη Print speed    |        |        |        |          |
| #1  | 1      | 0      | -1.757 | 10.04    |

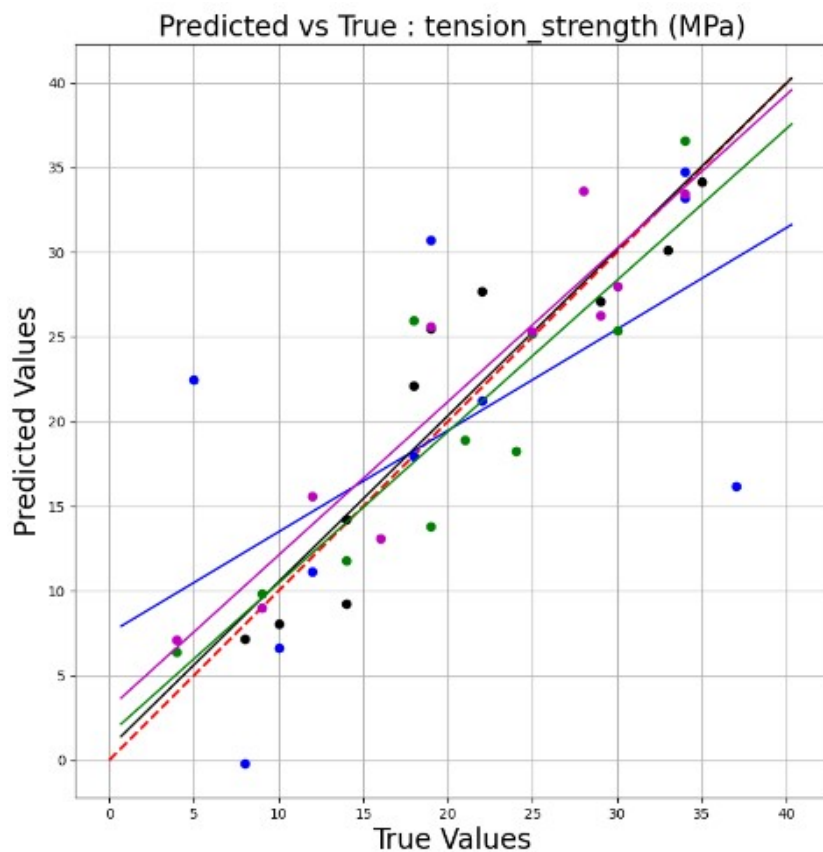
Πίνακας 8: Απόδοση νευρωνικού δικτύου εκτίμησης ορίου διαρροής με προσθήκη στις παραμέτρους το Infill Pattern και το Print speed

Συγκρίνοντας του Πίνακες 7, 8 η προσθήκη παραπάνω μεταβλητών εισόδου, δεν βελτιώνει την απόδοση του Ν.Δ. Επομένως το βέλτιστο Ν.Δ. είναι αυτό που υπολογίστηκε αρχικά. Η αρχιτεκτονική του Ν.Δ. φαίνεται παρακάτω



Σχήμα 10: Αρχιτεκτονική Νευρωνικού Δικτύου του Ορίου Διαρροής

Στο σχήμα 11 παρουσιάζεται το  $R^2$  plot. Πάλι, για την γενίκευση των αποτελεσμάτων, παρουσιάζεται το γράφημα αυτό για τέσσερα νευρωνικά δίκτυα ίδιας αρχιτεκτονικής ( Αρχιτεκτονική (4,13) ) τα οποία έχουν εκπαιδευτεί σε διαφορετικό υποσύνολο παραμέτρων train dataset και συνεπώς αξιολογούνται στο συμπληρωματικό (test dataset). Και σε αυτή την περίπτωση η κόκκινη διακεκομμένη γραμμή αντιστοιχεί στην γραμμή  $y = x$ , ενώ οι υπόλοιπες ευθείες έχουν προκύψει από γραμμική παλινδρόμηση των εκτιμώμενων και πραγματικών τιμών κάθε σετ αξιολόγησης. Παρατηρείται πως τα 3 dataset είχαν πολύ ικανοποιητική απόδοση και πλησιάζουν πολύ την ευθεία  $y = x$ . Αντίθετα το μπλε dataset δίνει κακές προβλέψεις. Αυτό οφείλεται στο γεγονός ότι το νευρωνικό δίκτυο είναι κάπως ευαίσθητο στα δεδομένα εισόδου, το οποίο είναι λογικό καθώς το υποσύνολο εκπαίδευσης είναι πολύ περιορισμένο.



Σχήμα 11:  $R^2$  γράφημα για το όριο διαρροής. Με κόκκινο είναι η γραμμή  $y = x$

#### 4.4 Βελτιστοποίηση Νευρωνικού δικτύου πρόβλεψης επιμήκυνσης

Αρχικά εκπαιδεύτηκε το Ν.Δ. με παραμέτρους εισόδους, που επιλέχθηκαν από τα στατιστικά τεστ συσχέτισης

$$arr = [1, 5, 8, 9] = [layer\_height, nozzle\_temperature, material, fan\_speed] \quad (5)$$

Στον παρακάτω πίνακα φαίνονται οι καλύτερες αρχιτεκτονικές κατά μέση τιμή, καθώς και η στατιστικά ισοδύναμη καλύτερη λύση.

| Solutions                                 | Layer1 | Layer2 | $\mu$ | $\sigma$ |
|---|--------|--------|-------|----------|
| <i>Κατάταξη κατά μέση τιμή</i>            |        |        |       |          |
| #1  | 37     | 0      | 0.562 | 0.00380  |
| #2  | 10     | 16     | 0.561 | 0.00552  |
| #3  | 23     | 0      | 0.558 | 0.00310  |
| #4  | 7      | 28     | 0.556 | 0.00310  |
| #5  | 25     | 16     | 0.551 | 0.00449  |
| <i>Στατιστικά ισοδύναμη καλύτερη λύση</i> |        |        |       |          |
| #1  | 7      | 0      | 0.548 | 0.0137   |

Πίνακας 9: Απόδοση νευρωνικού δικτύου εκτίμησης επιμήκυνσης με παραμέτρους εισόδου που αναφέρθηκαν

Στην συνέχεια, έγιναν επιπλέον τεστ βελτιστοποίησης, με επιπλέον εισόδους κάθε φορά τις παραμέτρους Wall Thickness το Infill Density το Infill Pattern και Print Speed. Οι παράμετροι αυτοί, υπενθυμίζεται πως δεν εξετάστηκαν αρχικά, καθώς από τον Πίνακα 4, οι τιμές MI είναι μηδενικές και οι τιμές Ftest σχετικά χαμηλές.

| Solutions  | Layer1 | Layer2 | $\mu$ | $\sigma$ |
|--|--------|--------|-------|----------|
| <i>Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη wall thickness</i> |        |        |       |          |
| #1   | 22     | 7      | 0.627 | 0.0174   |
| <i>Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη Infill Density</i> |        |        |       |          |
| #1   | 19     | 10     | 0.741 | 0.0229   |
| <i>Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη Infill Pattern</i> |        |        |       |          |
| #1   | 19     | 13     | 0.551 | 0.0377   |
| <i>Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη Print Speed</i>    |        |        |       |          |
| #1   | 23     | 0      | 0.517 | 0.0577   |

Πίνακας 10: Απόδοση νευρωνικού δικτύου εκτίμησης επιμήκυνσης με προσθήκη στις παραμέτρους εισόδου το Wall Thickness το Infill Density και το Infill Pattern

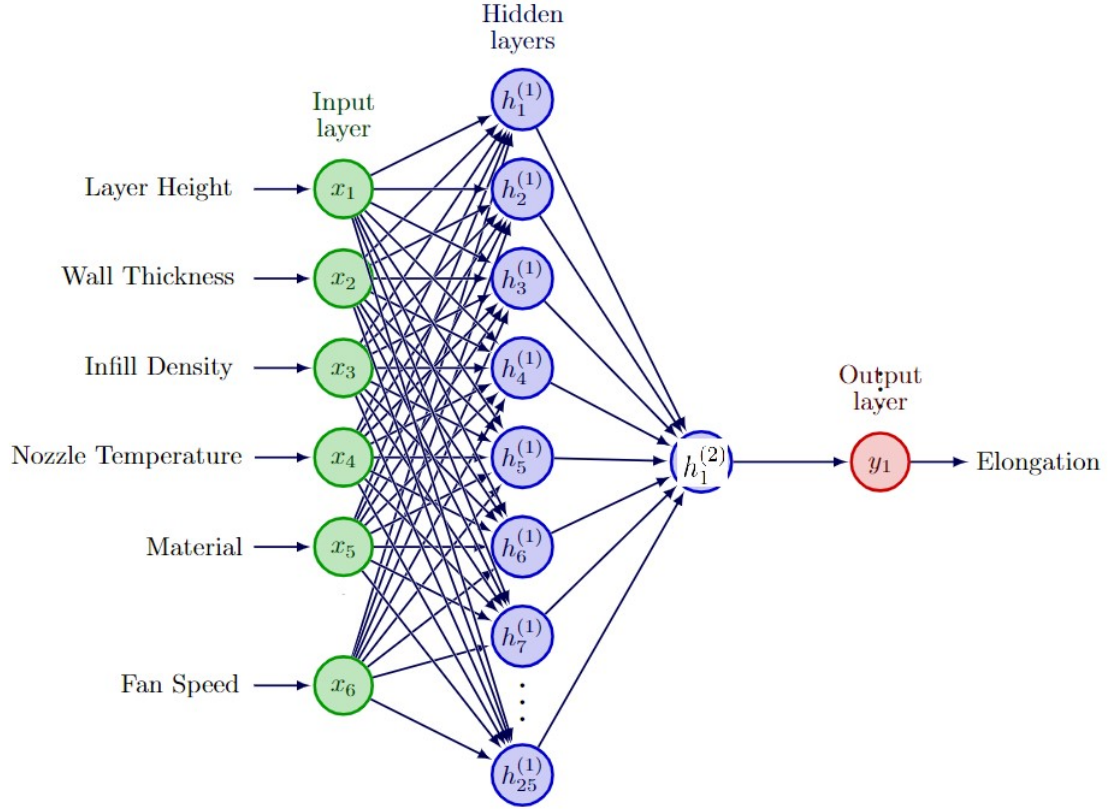
Συγκρίνοντας τους Πίνακες 9, 10, παρατηρείται πως η προσθήκη του Infill Density βελτιώνει σημαντικά την απόδοση του Ν.Δ., καθώς και η προσθήκη του Wall Thickness. Αντίθετα οι προσθήκη των άλλων παραμέτρων χειροτερεύει την απόδοση και για αυτό δεν εξετάζονται και περαιτέρω συνδυασμοί. Αξίζει σε αυτό το σημείο, να αξιολογήσουμε τα Ν.Δ. με την ταυτόχρονη προσθήκη των παραμέτρων Wall Thickness και Infill Density επομένως το διάγραμμα των μεταβλητών εισόδου είναι

$$arr = [1, 2, 3, 5, 8, 9] = [layer\_height, wall\_thickness, infill\_density, nozzle\_temperature, material, fan\_speed] \quad (6)$$

| Solutions  | Layer1 | Layer2 | $\mu$ | $\sigma$ |
|--|--------|--------|-------|----------|
| <i>Στατιστικά ισοδύναμη καλύτερη λύση με προσθήκη Wall Thickness, Infill Density</i> |        |        |       |          |
| #1   | 25     | 1      | 0.758 | 0.00929  |

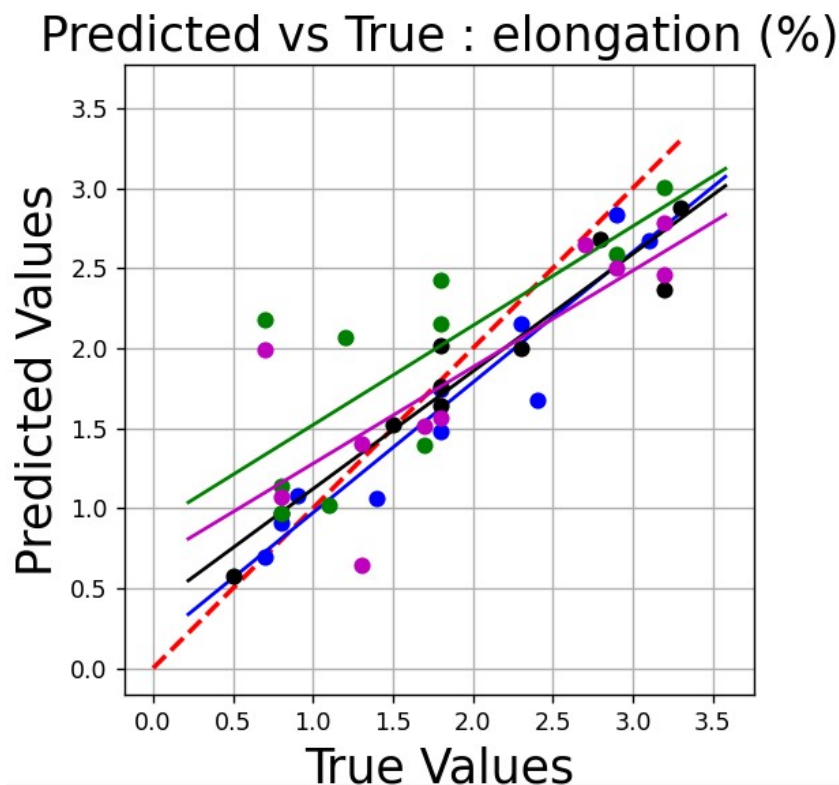
Πίνακας 11: Απόδοση νευρωνικού δικτύου εκτίμησης επιμήκυνσης με προσθήκη στις παραμέτρους εισόδου το Wall Thickness και το Infill Density

Συγκρίνοντας τους Πίνακες 10, 11 παρατηρείται ότι η μέση τιμή ( $\mu$ ) αξιολόγηση παραμένει σε παρόμοια επίπεδα, όπως και η τυπική απόκλιση ( $\sigma$ ). Στην πρώτη περίπτωση (Πίνακας 10) ο αριθμός των βαρών είναι ( $w_2 = 5x_{19} + 19x_{10} + 10x_1 = 295$ ), ενώ στην δεύτερη περίπτωση είναι (Πίνακας 11) ο αριθμός των βαρών είναι ( $w_3 = 6x_{25} + 25x_1 + 1x_1 = 176$ ). Για να αποφύγουμε την περίπτωση του Overfitting επιλέγουμε την περίπτωση όπου, τα βάρη είναι λιγότερα, επομένως γίνεται η προσθήκη του Wall Thickness και του Infill Density. Παρακάτω φαίνεται το σχήμα του Νευρωνικού Δικτύου.



Σχήμα 12: Αρχιτεκτονική Νευρωνικού Δικτύου της παραμόρφωσης

Στο σχήμα 13 παρουσιάζεται το  $R^2$  plot. Πάλι, για την γενίκευση των αποτελεσμάτων, παρουσιάζεται το γράφημα αυτό για τέσσερα νευρωνικά δίκτυα ίδιας αρχιτεκτονικής ( Αρχιτεκτονική (25,1) ) τα οποία έχουν εκπαιδευτεί σε διαφορετικό υποσύνολο παραμέτρων train dataset και συνεπώς αξιολογούνται στο συμπληρωματικό (test dataset). Και σε αυτή την περίπτωση η κόκκινη διακεκομμένη γραμμή αντιστοιχεί στην γραμμή  $y = x$ , ενώ οι υπόλοιπες ευθείες έχουν προκύψει από γραμμική παλινδρόμηση των εκτιμώμενων και πραγματικών τιμών κάθε σετ αξιολόγησης. Παρατηρείται πως οι ευθείες που προκύπτουν με γραμμική παλινδρόμηση έχουν μικρότερη κλίση από 1, που σημαίνει πως υπερεκτιμούν τις χαμηλές τιμές παραμόρφωσης ενώ υποεκτιμούν τις υψηλές τιμές της παραμόρφωσης. Το συγκεκριμένο χαρακτηριστικό ίσως χρειάζεται περισσότερα δεδομένα-τεστς ώστε το νευρωνικό δίκτυο να προσαρμοστεί καλύτερα.



Σχήμα 13:  $R^2$  γράφημα για την παραμόρφωση. Με κόκκινο είναι η γραμμή  $y = x$



## 5. Αναφορές

- [1] Mohammad S. Alsoufi and Abdulrhman E. Elsayed. How surface roughness performance of printed parts manufactured by desktop fdm 3d printer with pla+ is influenced by measuring direction. *American Journal of Mechanical Engineering*, 5(5):211–222, 2017.
- [2] Christin Arnold, Delf Monsees, Jeremias Hey, and Ramona Schweyen. Surface quality of 3d-printed models as a function of various printing parameters. *Materials (Basel)*, 12(12):1970, June 2019.
- [3] Irene Buj-Corral, Xavier Sánchez-Casas, and Carmelo J. Luis-Pérez. Analysis of am parameters on surface roughness obtained in pla parts printed with fff technology. *Polymers*, 13(14), 2021.
- [4] Edwin-Joffrey Courtial, Clément Perrinet, Arthur Colly, David Mariot, Jean-Marc Frances, René Fulchiron, and Christophe Marquette. Silicone rheological behavior modification for 3d printing: Evaluation of yield stress impact on printed object properties. *Additive Manufacturing*, 28:50–57, 2019.
- [5] Damir Godec, Santiago Cano, Clemens Holzer, and Joamin Gonzalez-Gutierrez. Optimization of the 3d printing parameters for tensile properties of specimens produced by fused filament fabrication of 17-4ph stainless steel. *Materials*, 13(3), 2020.
- [6] Marco Leite, João Fernandes, Augusto Deus, Luis Reis, and M.F. Vaz. Study of the influence of 3d printing parameters on the mechanical properties of pla. 05 2018.
- [7] J.A. Travieso-Rodriguez, Ramon Jerez-Mesa, Jordi Llumà, Oriol Traver, Giovanni Gómez-Gras, Joan Josep, and Roa Rovira. Mechanical properties of 3d-printing polylactic acid parts subjected to bending stress and fatigue testing. *Materials*, 2019:3859, 11 2019.

## 6. Κώδικες

Παρουσιάζεται ο γενικός κώδικας:

```
1 import numpy as np
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.neural_network import MLPRegressor
5 from sklearn.preprocessing import StandardScaler
6 import matplotlib.pyplot as plt
7 from scipy import stats as st
8
9
10 scalerX = StandardScaler()
11 df = pd.read_csv("Data.csv")
12
13 #Targets = ['roughness ( \mu m)', 'tension_strength (MPa)', 'elongation (%)']
14 Targets = df.columns[-3:]
15 ystring = Targets[0]
16
17 #Number of trials -> statistical analysis
18 ki=8
19 r2_table = np.zeros(ki) #save the r2 value for statistical test
20
21 #Check alternative models
22 nkTable = [1]; specific = 1
23
24
25 #nkTable = [2,7]; specific = 0 #Roughness
26 #nkTable = [3,6]; specific = 0 #uts
27 #nkTable = [2]; specific = 0 #elongation , 3
28
29 Final = np.zeros([5*nkTable._len_,4])
30 FinalST = np.zeros([ nkTable._len_,4])
31 ali = -1 #alternatives_index
32
33 for nk in nkTable:# specific
34     ali +=1
35
36     if ystring == Targets[0]:
37         if specific==1:
38             #—— roughness
39             arr = [0,4,6]
40         else:
41             arr = [0,nk,6]
42     if ystring == Targets[1]:
43         if specific==1:
44             #—— MPA
45             arr = [0,1,2,4,7,8] #-> new = better
46         else:
47             arr = [0,1,2,nk,4,7,8] #3 -> bibliography , 6 -> bibliography + ftest
48     if ystring == Targets[2]:
49         if specific==1:
50             #—— elognation
51             arr = [0,1,4,7,8]
52         else:
53             arr = [0,1,nk,4,7,8] # 2-> fi test -> 2->0.78
54
```

```

55 #Best, independant of nn architecture
56 #region
57 cn=0
58 MATRIX=[]
59 OBJ=[]
60 #maximum 2 layers
61 for i in range(1,51,2):
62     MATRIX.append([i, 0])
63     cn=cn+1
64 for i in range(1,31,3):
65     for j in range(1,31,3):
66         MATRIX.append([i, j])
67         cn=cn+1
68
69 OBJ = np.zeros(cn) #score
70 SD  = np.zeros(cn) #variance of score
71
72 #endregion
73
74 u = cn
75 for j in range(0,u):
76
77     print(j)
78     R2=0
79     for k in range(ki):
80         train, test = train_test_split(df, test_size=0.2) #different data
81         X_train = train.iloc[:,arr]
82         Y_train = train[ystring]
83         X_test  = test.iloc[:,arr]
84         Y_test  = test[ystring]
85
86         #Data Preprocess
87         scaler = StandardScaler()
88         scaler.fit(X_train)
89
90         X_train = scaler.transform(X_train)
91         X_test  = scaler.transform(X_test)
92
93         #Regression
94         if MATRIX[j][1] == 0:
95             MLP=MLPRegressor(hidden_layer_sizes=(MATRIX[j][0]), max_iter=5000, random_state
96                             =1)
97         else:
98             MLP=MLPRegressor(hidden_layer_sizes=(MATRIX[j][0],MATRIX[j][1]), max_iter=5000,
99                             random_state=1)
100
101         MLP.fit(X_train, Y_train.values.ravel())
102
103         #R2 plot, scaled to 0-1
104         y_pred = MLP.predict(X_test)
105         y_true = Y_test.values
106
107         y_tmax = y_true.max()
108         y_pmax = y_pred.max()
109
110         r2 = MLP.score(X_test, Y_test)
111         R2=R2+r2/ki

```

```

110     print(r2)
111     #Statistical Tets
112     r2_table[k] = r2
113
114
115     OBJ[j] = R2
116     SD[j] = r2_table.var()
117
118     print('R2: {:.4f}'.format(R2))
119     print('SD: {:.4f}'.format(SD[j]))
120
121     #save all data
122     TXT= np.zeros([cn,4])
123     for j in range(0,u):
124         TXT[j,:] = [MATRIX[j][0],MATRIX[j][1],OBJ[j],SD[j]]
125     #print(TXT)
126
127     #Show best 5
128     id = np.argsort(OBJ)
129     # print(id[-5:]+1)
130     print( TXT[id[-5:],:] )
131     Final[5*ali:(5*ali+5),:] = TXT[id[-5:],:]
132     # print(Final)
133
134     #statistic test
135     #st_id = [0,0,0,0,0]
136
137     best = 0
138     for j in range(1,u):
139         mu1 = TXT[best,2]
140         mu2 = TXT[j,2]
141         s1_2 = TXT[best,3]
142         s2_2 = TXT[j,3]
143
144         if ( st.ttest_ind_from_stats(mu2, s2_2 ,ki, mu1, s1_2, ki, equal_var=False,
145             alternative='greater').pvalue < 0.15 ):
146             #pvalue < 0.05 ->
147             #the propability that mu2 > mu1 covers and area 1-pvalue
148             #So if i want to be 95% sure that mu2>mu1, that means that:
149             #area > 0.95 -> 1-pvalue >0.95 -> pvalue <0.05
150
151             best = j
152             print("Best is : {:.d}".format(best))
153
154     FinalST[ali,:] = TXT[best,:]
155
156     print(FinalST)
157
158     print("Final best 5 table:")
159     print(Final)

```

Παρουσιάζεται ο κώδικας υπολογισμού των στατιστικών δεικτών *f<sub>test</sub>*, *mi*:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.decomposition import PCA
6
7 scalerX = StandardScaler()
8
9 #flags
10 printData = 0
11 selectY = 1 #0 for roughness, 1 for UTS, 2 for elongation
12 selectByF = 1
13 ki = 10
14
15 miTotal = np.zeros([ki,9])
16 FiTotal = np.zeros([ki,9])
17
18
19 miT = np.zeros(9)
20 FiT = np.zeros(9)
21
22 #####-----Data Import and preprocessing
23 #region
24 #-----Import
25 Data = pd.read_excel('Data.xlsx')
26 #print(Data.head())
27 # for col in Data.columns:
28 #     print(col)
29
30 # 3 targets.
31 nset = len(Data.columns)- 3
32
33 #-----Convert categorical to numeric
34 StringArray = Data.dtypes == object
35 StringArray = 1*StringArray._.array_()
36 #print(StringArray)
37
38 #infill pattern
39 DifferentPatterns = Data['infill_pattern'].unique()
40 nPatterns = len( DifferentPatterns )
41 Data['infill_pattern'].replace(DifferentPatterns ,range(nPatterns),inplace = True)
42 #Data["infill_pattern"].replace(['grid','honeycomb'],[0,1],inplace=True)
43
44 #print(Data['infill_pattern'])
45
46 #materials
47 DifferentMaterials = Data['material'].unique()
48 nMaterials = len( DifferentMaterials )
49 #not the best way, if there were 3 categories, new features should be introduced
50 Data['material'].replace(DifferentMaterials ,range(nMaterials),inplace = True)
51
52 #print(Data['material'])
53
54 #-----Seperate Matrices
55 #Target
56 Y_df = Data.iloc[:,nset+selectY]
```

```

57 #print(Y_df)
58
59 #drop other outputs
60 X_df = Data.drop(Data.columns[[9,10,11]], axis=1)
61 StringArray = np.delete(StringArray, range(nset,12))
62
63 Xnames= Data.columns
64 #print(X_df.head())
65
66
67 #-----Make nd arrays
68 X = X_df.to_numpy()
69 Y = Y_df.to_numpy()
70
71 np.set_printoptions(formatter={'float': lambda x: "{0:0.3f}".format(x)})
72
73 for k in range(ki):
74
75     # #-----F test
76     from sklearn.feature_selection import SelectPercentile, f_regression,
        mutual_info_regression
77
78     fi, _ = f_regression(X,Y)
79     mi = mutual_info_regression(X,Y)
80
81
82     FiT = FiT + fi/ki
83     miT = miT + mi/ki
84
85     miTotal[k,:] = mi
86     FiTotal[k,:] = fi
87
88
89 print("Mutual Info Regression Total: ")
90 print(miTotal)
91 print('_____')
92 print("Mutual Information Average: ")
93 print(miT)
94
95 print('_____')
96 print('_____')
97 print("F regression Total: ")
98 print(FiTotal)
99 print('_____')
100 print("F regression Average: ")
101 print(FiT)

```

Παρουσιάζεται ο κώδικας για την δημιουργία των  $R^2$  γραφημάτων:

```
1 import numpy as np
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.neural_network import MLPRegressor
5 from sklearn.preprocessing import StandardScaler
6 import matplotlib.pyplot as plt
7 from scipy import stats as st
8 from sklearn.linear_model import LinearRegression
9
10 np.set_printoptions(formatter={'float': lambda x: "{0:0.3f}".format(x)})
11
12 scalerX = StandardScaler()
13 df = pd.read_csv("Data.csv")
14
15 #Targets = ['roughness (\mu m)', 'tension_strength (MPa)', 'elongation (%)']
16 Targets = df.columns[-3:]
17 ystring = Targets[0]
18
19 #Number of trials -> statistical analysis
20 ki=8
21 r2_table = np.zeros(ki) #save the r2 value for statistical test
22 printr2 = 1 #r2 graph
23
24
25 #-----Check alternative models
26 nkTable = [1]; specific = 1
27 #arr = [0,1,2,4,7,8] # elongation
28 #arr = [0,1,2,4,7,8] # -> MPA
29 arr = [0,4,6] #-> roughness
30
31 #put neural network architecture
32 MATRIX = [[7,10]] #roughness
33 #MATRIX = [[4,13]] #uts
34 #MATRIX = [[25,1]] #elongation
35
36 Final = np.zeros([ nkTable._len_(),4])
37 FinalST = np.zeros([ nkTable._len_(),4])
38
39
40 xx = np.zeros(ki*10)
41 yy = np.zeros(ki*10)
42 coeff = np.zeros(ki)
43 inter = np.zeros(ki)
44
45
46
47 j = 0 #in order to use older code, where there were a lot of combinations, we use j var
48 R2=0
49 for k in range(ki):
50     train, test = train_test_split(df, test_size=0.2, random_state=k) #different data
51     X_train = train.iloc[:,arr]
52     Y_train = train[ystring]
53     X_test = test.iloc[:,arr]
54     Y_test = test[ystring]
55
56     #Data Preprocess
```

```

57     scaler = StandardScaler()
58     scaler.fit(X_train)
59
60     X_train = scaler.transform(X_train)
61     X_test = scaler.transform(X_test)
62
63     #Regression
64     if MATRIX[j][1] == 0:
65         MLP=MLPRegressor(hidden_layer_sizes=(MATRIX[j][0]), max_iter=5000,random_state=1)
66     else:
67         MLP=MLPRegressor(hidden_layer_sizes=(MATRIX[j][0],MATRIX[j][1]), max_iter=5000,
68                               random_state=1)
69
70     MLP.fit(X_train, Y_train.values.ravel())
71
72     r2 = MLP.score(X_test, Y_test)
73     R2=R2+r2/ki
74     print(r2)
75
76     #For variance
77     r2_table[k] = r2
78
79     #R2 plot
80     y_pred = MLP.predict(X_test)
81     y_true = Y_test.values
82
83     #Regression for r2
84     reg = LinearRegression().fit(y_true.reshape(-1, 1), y_pred.reshape(-1, 1))
85
86     coeff[k] = reg.coef_
87     inter[k] = reg.intercept_
88
89     xx[10*k:10*(k+1)] = y_true
90     yy[10*k:10*(k+1)] = y_pred
91
92     SD = r2_table.var()
93
94     print('R2: {:.4f}'.format(R2))
95     print('SD: {:.4f}'.format(SD))
96
97     # Print Results
98     RESULTS = [MATRIX[0][0],MATRIX[0][1],R2,SD]
99     print("Results: ")
100     print(RESULTS)
101
102     #print calibration
103     xmax = xx.max()
104     xmin = xx.min()
105     lx = 0.1*(xmax - xmin)
106
107     ymax = yy.max()
108     ymin = yy.min()
109     ly = 0.1*(ymax - ymin)
110
111     #plot only 4, in order for the graph to be clear
112     colors = ['b','k','g','m']
113     if print_r2:

```



```

113 plt.figure(1)
114 plt.plot([0,ymax+ly], [0,ymax+ly], 'r—',linewidth = '2',label = 'y=x')
115 for k in range(4):
116
117     a = coeff[k]
118     b = inter[k]
119
120     plt.plot(xx[10*k:10*(k+1)], yy[10*k:10*(k+1)], 'o',color=colors[k] )
121     plt.plot(np.array([xmin - lx,xmax+lx]),a*np.array([xmin - lx,xmax+lx])+b,'—',color=
122             colors[k] )
123
124 plt.xlabel('True Values',fontsize=20)
125 plt.ylabel('Predicted Values',fontsize=20)
126 titletxt = 'Predicted vs True : {s}'.format(ystring)
127 plt.title(titletxt, fontsize =20)
128 #plt.ylim(ymin - ly,ymax+ly)
129 plt.ylim(0,ymax+ly)
130 plt.xlim(0,xmax+lx)
131 plt.axis('square')
132 #plt.legend(handles=[line1])
133 plt.grid()
plt.show()

```