



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΜΕΤΑΦΟΡΩΝ ΚΑΙ ΣΥΓΚΟΙΝΩΝΙΑΚΗΣ ΥΠΟΔΟΜΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΟΔΟΣΤΡΩΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΠΡΟΒΛΕΨΗ ΟΜΑΛΟΤΗΤΑΣ ΟΔΟΣΤΡΩΜΑΤΟΣ ΜΕ ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ
ΔΙΚΤΥΩΝ**



ΑΣΩΝΙΤΗ ΔΗΜΗΤΡΑ

Επιβλέπουσα: Χριστίνα Πλατή, Αναπληρώτρια Καθηγήτρια

ΜΑΡΤΙΟΣ 2023

ΕΥΧΑΡΙΣΤΙΕΣ

Πρώτα από όλα θα ήθελα να ευχαριστήσω την Αναπληρώτρια Καθηγήτρια της Σχολής των Πολιτικών Μηχανικών κ. Χριστίνα Πλατή, για την ευκαιρία που μου έδωσε να ασχοληθώ με το παρόν θέμα, αλλά και την πολύτιμη βοήθεια και καθοδήγηση καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Στη συνέχεια, θα ήθελα να ευχαριστήσω τον κ. Μύρων Σαμψάκη για την αμέριστη βοήθεια, την προθυμία του και τον χρόνο που κατέβαλε για να με καθοδηγήσει και να με συμβουλέψει στο κομμάτι της ανάλυσης των Νευρωνικών Δικτύων, της παρούσας διπλωματικής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου, για την κατανόηση, την υπομονή και τη συμπαράσταση που μου έδειξαν όλα αυτά τα χρόνια.

ΠΕΡΙΛΗΨΗ

Στο πλαίσιο διαχείρισης και συντήρησης των οδοστρωμάτων, απαραίτητη είναι η εκτίμηση και αξιολόγηση διαφόρων παραμέτρων. Μία από αυτές τις παραμέτρους είναι η ομαλότητα του οδοστρώματος. Τα στοιχεία κυκλοφορίας, οι καιρικές και περιβαλλοντικές συνθήκες, καθώς και η δομική κατάσταση του οδοστρώματος παίζουν καθοριστικό ρόλο στη μείωση της ομαλότητας. Για την αξιολόγηση της ομαλότητας, χρησιμοποιούνται διάφοροι δείκτες, με πιο γνωστό τον πλέον ευρέως διαδεδομένο δείκτη IRI (International Roughness Index), ενώ έχουν γίνει πολλές έρευνες για την ανάπτυξη μοντέλων πρόβλεψης της ομαλότητας. Από την άλλη μεριά, τα Νευρωνικά Δίκτυα καλούνται να φέρουν λύση σε πολύπλοκα προβλήματα, αγνοώντας τους θεμελιώδεις νόμους της φυσικής προκειμένου να αποδίδουν καλά στις παρατηρήσεις. Ένα από αυτά τα προβλήματα είναι και η πρόβλεψη της ομαλότητας των οδοστρωμάτων. Τα Νευρωνικά Δίκτυα αποτελούν ένα είδος μηχανικής μάθησης. Η κατασκευή τους βασίζεται στην κατανόηση της λειτουργίας του ανθρώπινου εγκεφάλου, από όπου και πήραν το όνομά τους. Τις τελευταίες δεκαετίες, έχουν διεξαχθεί πολλές έρευνες για την πρόβλεψη της ομαλότητας των οδοστρωμάτων, με τη χρήση Νευρωνικών Δικτύων, στο πλαίσιο των οποίων εντάσσεται και η παρούσα διπλωματική εργασία. Αποδεικνύεται ότι τα Νευρωνικά Δίκτυα, μέσω των προβλέψεων των μελλοντικών τιμών του δείκτη ομαλότητας IRI, είναι ένα χρήσιμο εργαλείο για το μηχανικό, καθώς υποστηρίζει τον έγκαιρο προγραμματισμό και τη λήψη μέτρων συντήρησης του οδοστρώματος με στόχο την ελαχιστοποίηση του χρόνου επέμβασης, αλλά και του σχετικού κόστους.

Λέξεις κλειδιά: Νευρωνικά Δίκτυα, ομαλότητα, IRI, πρόβλεψη, MLP, LSTM

ABSTRACT

In the context of road surface management and maintenance, it is necessary to assess and evaluate various parameters. One of these parameters is the roughness of the road surface. Traffic data, weather and environmental conditions, as well as the structural condition of the road surface play a crucial role in reducing normality. To assess normality, various indicators are used, the most widely known being the most widely used IRI (International Roughness Index), and many researches have been done to develop models for predicting normality. On the other hand, Neural Networks are called upon to provide a solution to complex problems, ignoring the fundamental laws of physics in order to perform well in observations. One of these problems is the prediction of the roughness of the road surfaces. Neural Networks are a kind of machine learning. Their construction is based on an understanding of the functioning of the human brain, from which they got their name. In recent decades, many researches have been carried out to predict the roughness of road surfaces, using Neural Networks, within the framework of which this diploma thesis is included. It is proven that Neural Networks, through the predictions of future values of the IRI roughness index, is a useful tool for the engineer, as it supports the timely planning and taking of road maintenance measures in order to minimize the intervention time, but also the associated costs.

Key words: Neural Networks, roughness, IRI, prediction, MLP, LSTM

ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΕΙΣΑΓΩΓΗ	1
1.1	Αντικείμενο.....	1
1.2	Στόχος και μεθοδολογία.....	2
1.3	Δομή εργασίας	3
2.	ΟΜΑΛΟΤΗΤΑ ΟΔΟΣΤΡΩΜΑΤΟΣ	4
2.1	Ορισμός και παράγοντες επιρροής.....	4
2.2	Δείκτες έκφρασης ομαλότητας	6
2.3	Διεθνής δείκτης ομαλότητας (IRI)	6
2.3.1	Γενικά.....	6
2.3.2	Πλεονεκτήματα	7
2.3.3	Όρια προδιαγραφών	8
2.4	Συστήματα καταγραφής.....	9
3.	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	11
3.1	Εισαγωγή στα Νευρωνικά Δίκτυα	11
3.1.1	Γενικά.....	11
3.1.2	Ορισμός	12
3.1.3	Ιστορική Αναδρομή	12
3.1.4	Πλεονεκτήματα Νευρωνικών Δικτύων.....	14
3.2	Δομή του νευρώνα	14
3.3	Αρχιτεκτονική Νευρωνικών Δικτύων	15
3.3.1	Βασικές κατηγορίες.....	15
3.3.2	Multilayer Perceptron (MLP).....	16
3.3.3	Recurrent Neural Network (RNN)	17
3.3.4	Long Short Term Memory (LSTM)	17
3.3.5	Convolutional Neural Network (CNN)	18
3.4	Τεχνικές εκπαίδευσης	20
3.4.1	Γενικά.....	20
3.4.2	Τεχνική παλινδρόμησης (regression)	20
3.4.2.1	Γενικά	20
3.4.2.2	Linear Regression	22
3.4.2.3	Ridge Regression	24
3.4.2.4	Lasso Regression.....	26

3.4.3	Τεχνική ταξινόμησης (classification)	26
4.	ΑΝΑΛΥΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ.....	29
4.1	Συνοπτική περιγραφή	29
4.2	Συλλογή στοιχείων	30
4.2.1	Δείκτες ομαλότητας IRI	30
4.2.2	Πάχη οδοστρώματος	31
4.2.3	Στοιχεία κυκλοφορίας	33
4.3	Πλαίσιο πειραματικών αναλύσεων	35
4.4	Επιλογή Μοντέλου	36
4.5	Γραμμικές συναρτήσεις.....	37
4.5.1	Linear Regression	39
4.5.2	Ridge Regression	40
4.5.3	Lasso Regression.....	41
4.6	Εφαρμογή Νευρωνικών Δικτύων	41
4.6.1	MLP.....	41
4.6.1.1	Εισαγωγή και διάταξη των δεδομένων στο δίκτυο.....	42
4.6.1.2	Παραμετροποίηση της αρχιτεκτονικής του δικτύου.....	43
4.6.1.3	Κατασκευή της συνάρτησης εκπαίδευσης.....	44
4.6.2	LSTM	49
4.6.2.1	Εισαγωγή και διάταξη των δεδομένων στο δίκτυο.....	50
4.6.2.2	Παραμετροποίηση της αρχιτεκτονικής του δικτύου.....	51
4.6.2.3	Κατασκευή της συνάρτησης εκπαίδευσης.....	52
5.	Αποτελέσματα.....	56
5.1	Αποτελέσματα 1 ^{ης} πειραματικής ανάλυσης	56
5.2	Αποτελέσματα 2 ^{ης} πειραματικής ανάλυσης	57
5.3	Αποτελέσματα 3 ^{ης} πειραματικής ανάλυσης	59
5.4	Αποτελέσματα 4 ^{ης} πειραματικής ανάλυσης	60
5.5	Συγκριτική ανάλυση	62
6.	ΣΥΜΠΕΡΑΣΜΑΤΑ	65
7.	Βιβλιογραφία	67

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 2.1: ΕΠΙΦΑΝΕΙΑ ΟΔΟΣΤΡΩΜΑΤΟΣ ΚΑΙ ΚΑΤΑΝΟΜΕΣ (ΠΡΟΦΙΛ)	4
ΕΙΚΟΝΑ 2.2: ΣΤΟΙΧΕΙΑ ΑΣΤΟΧΙΩΝ ΕΠΙΠΕΔΟΤΗΤΑΣ.....	5
ΕΙΚΟΝΑ 2.3: ΠΡΟΣΟΜΟΙΩΜΑ ΕΝΟΣ ΤΕΤΑΡΤΟΥ ΤΟΥ ΟΧΗΜΑΤΟΣ	7
ΕΙΚΟΝΑ 3.1: ΔΟΜΗ ΝΕΥΡΩΝΑ	15
ΕΙΚΟΝΑ 3.2: ΑΡΧΙΤΕΚΤΟΝΙΚΗ MULTILAYER PERCEPTRON (MLP).....	16
ΕΙΚΟΝΑ 3.3: ΑΡΧΙΤΕΚΤΟΝΙΚΗ LSTM RNN	17
ΕΙΚΟΝΑ 3.4: ΠΑΡΟΥΣΙΑΣΗ ΑΡΙΘΜΗΤΙΚΩΝ ΠΡΑΞΕΩΝ ΣΕ ΕΝΑ ΚΡΥΦΟ ΕΠΙΠΕΔΟ LSTM ΜΕ ΔΥΟ ΠΥΛΕΣ ΕΙΣΟΔΟΥ	18
ΕΙΚΟΝΑ 3.5: ΠΑΡΑΔΕΙΓΜΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ CNN	19
ΕΙΚΟΝΑ 3.6: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΜΟΝΤΕΛΟΥ ΜΕ ΤΗ ΜΕΘΟΔΟ ΤΗΣ ΠΑΛΙΝΔΡΟΜΗΣΗΣ	22
ΕΙΚΟΝΑ 3.7: ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΜΟΝΤΕΛΟΥ ΜΕ ΤΗ ΜΕΘΟΔΟ ΤΗΣ ΤΑΞΙΝΟΜΗΣΗΣ	27
ΕΙΚΟΝΑ 4.1: ΟΡΙΣΜΟΣ ΠΕΙΡΑΜΑΤΙΚΩΝ ΑΝΑΛΥΣΕΩΝ	35
ΕΙΚΟΝΑ 4.2: ΔΙΑΧΩΡΙΣΜΟΣ ΔΕΔΟΜΕΝΩΝ ΣΕ ΔΕΔΟΜΕΝΑ TRAIN ΚΑΙ TEST	38
ΕΙΚΟΝΑ 4.3: ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΣΕ ΠΙΝΑΚΕΣ.....	39
ΕΙΚΟΝΑ 4.4: ΠΡΟΒΛΕΨΗ ΔΕΙΚΤΩΝ ΜΕΣΩ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ LINEAR REGRESSION	39
ΕΙΚΟΝΑ 4.5: ΠΡΟΒΛΕΨΗ ΔΕΙΚΤΩΝ ΜΕΣΩ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ RIDGE REGRESSION	40
ΕΙΚΟΝΑ 4.6: ΠΡΟΒΛΕΨΗ ΔΕΙΚΤΩΝ ΜΕΣΩ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ LASSO REGRESSION	41
ΕΙΚΟΝΑ 4.7: ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ MPL (DATASET).....	42
ΕΙΚΟΝΑ 4.8: ΟΡΙΣΜΟΣ ΤΟΥ ΑΡΙΘΜΟΥ ΔΕΙΓΜΑΤΟΣ ΚΑΙ ΕΠΙΛΟΓΗ ΤΥΧΑΙΩΝ ΔΕΔΟΜΕΝΩΝ	43
ΕΙΚΟΝΑ 4.9: ΕΠΙΠΕΔΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ MLP	43
ΕΙΚΟΝΑ 4.10: ΑΡΧΙΤΕΚΤΟΝΙΚΗ MLP	44
ΕΙΚΟΝΑ 4.11: ΔΙΑΧΩΡΙΣΜΩΝ TRAIN DATASET ΚΑΙ TEST DATASET ΣΤΟ MLP	45
ΕΙΚΟΝΑ 4.12: ΕΙΣΑΓΩΓΗ MLP ΚΑΙ ΚΑΘΟΡΙΣΜΟΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΣΦΑΛΜΑΤΟΣ.....	46
ΕΙΚΟΝΑ 4.13: ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΣΦΑΛΜΑΤΟΣ ΣΕ ΚΑΘΕ ΕΠΑΝΑΛΗΨΗ ΣΤΟ MLP .	47
ΕΙΚΟΝΑ 4.14: ΕΠΑΛΗΘΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΔΙΑΚΟΠΗ ΕΚΠΑΙΔΕΥΣΗΣ MLP	48
ΕΙΚΟΝΑ 4.15: ΤΕΛΟΣ ΕΚΠΑΙΔΕΥΣΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΒΕΛΤΙΣΤΟΥ ΜΟΝΤΕΛΟΥ ΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ MLP.....	49
ΕΙΚΟΝΑ 4.16: ΕΙΣΑΓΩΓΗ ΚΑΙ ΔΙΑΤΑΞΗ ΔΕΔΟΜΕΝΩΝ.....	50
ΕΙΚΟΝΑ 4.17: ΕΙΣΑΓΩΓΗ ΤΩΝ ΕΠΙΠΕΔΩΝ ΤΟΥ ΔΙΚΤΥΟΥ LSTM	51
ΕΙΚΟΝΑ 4.18: ΔΟΜΗ ΤΟΥ ΔΙΚΤΥΟΥ LSTM	52
ΕΙΚΟΝΑ 4.19: ΔΙΑΧΩΡΙΣΜΩΝ TRAIN DATASET ΚΑΙ TEST DATASET ΣΤΟ LSTM.....	53
ΕΙΚΟΝΑ 4.20: ΕΙΣΑΓΩΓΗ LSTM ΚΑΙ ΚΑΘΟΡΙΣΜΟΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΣΦΑΛΜΑΤΟΣ	53
ΕΙΚΟΝΑ 4.21: ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΣΦΑΛΜΑΤΟΣ ΣΕ ΚΑΘΕ ΕΠΑΝΑΛΗΨΗ ΣΤΟ LSTM	54
ΕΙΚΟΝΑ 4.22: ΕΠΑΛΗΘΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΔΙΑΚΟΠΗ ΕΚΠΑΙΔΕΥΣΗΣ LSTM.....	54
ΕΙΚΟΝΑ 4.23: ΤΕΛΟΣ ΕΚΠΑΙΔΕΥΣΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΒΕΛΤΙΣΤΟΥ ΜΟΝΤΕΛΟΥ ΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ LSTM	55
ΕΙΚΟΝΑ 5.1: ΣΦΑΛΜΑΤΑ ΜΕΘΟΔΩΝ ΠΡΟΒΛΕΨΗΣ ΓΙΑ ΤΗΝ 1 ^η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ.....	56
ΕΙΚΟΝΑ 5.2: ΣΦΑΛΜΑΤΑ ΜΕΘΟΔΩΝ ΠΡΟΒΛΕΨΗΣ ΓΙΑ ΤΗ 2 ^η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ.....	57
ΕΙΚΟΝΑ 5.3: ΣΦΑΛΜΑΤΑ ΜΕΘΟΔΩΝ ΠΡΟΒΛΕΨΗΣ ΓΙΑ ΤΗΝ 3 ^η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ.....	59
ΕΙΚΟΝΑ 5.4: ΣΦΑΛΜΑΤΑ ΜΕΘΟΔΩΝ ΠΡΟΒΛΕΨΗΣ ΓΙΑ ΤΗΝ 4 ^η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ.....	60

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 2.1: ΕΠΙΘΥΜΗΤΑ ΟΡΙΑ ΔΕΙΚΤΩΝ ΟΜΑΛΟΤΗΤΑΣ IRI ΣΕ ΟΔΟΣΤΡΩΜΑΤΑ ΜΕ ΔΙΑΦΟΡΕΤΙΚΑ ΟΡΙΑ ΤΑΧΥΤΗΤΩΝ.....	9
ΠΙΝΑΚΑΣ 4.1: ΕΜΗΚ ΤΩΝ ΤΜΗΜΑΤΩΝ Α ΚΑΙ Β.....	33
ΠΙΝΑΚΑΣ 4.2: ΙΤΑ ΑΝΑ ΛΩΡΙΔΑ ΓΙΑ ΤΑ ΤΜΗΜΑΤΑ Α ΚΑΙ Β	34

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

ΣΧΗΜΑ 2.1: ΔΙΑΒΑΘΜΙΣΗ ΔΕΙΚΤΩΝ IRI ΓΙΑ ΟΔΟΣΤΡΩΜΑΤΑ ΜΕ ΔΙΑΦΟΡΕΤΙΚΑ ΟΡΙΑ ΤΑΧΥΤΗΤΩΝ	8
ΣΧΗΜΑ 3.1: ΠΑΡΑΔΕΙΓΜΑ LINEAR REGRESSION ΜΕ ΜΙΑ ΑΝΕΞΑΡΤΗΤΗ ΜΕΤΑΒΛΗΤΗ	23
ΣΧΗΜΑ 3.2: RIDGE REGRESSION ΓΙΑ ΔΙΑΦΟΡΕΣ ΤΙΜΕΣ ΤΟΥ λ	25
ΣΧΗΜΑ 3.3: ΜΟΝΤΕΛΟ ΜΕ ΥΠΕΡΠΡΟΣΑΡΜΟΓΗ (OVERFITTING)	25
ΣΧΗΜΑ 4.1: ΔΕΙΚΤΕΣ IRI ΓΙΑ ΤΟ ΔΕΞΙ ΙΧΝΟΣ ΤΗΣ ΑΡΙΣΤΕΡΗΣ ΛΩΡΙΔΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ Α	30
ΣΧΗΜΑ 4.2: ΔΕΙΚΤΕΣ IRI ΓΙΑ ΤΟ ΔΕΞΙ ΙΧΝΟΣ ΤΗΣ ΑΡΙΣΤΕΡΗΣ ΛΩΡΙΔΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ Β.....	31
ΣΧΗΜΑ 4.3: ΠΑΧΗ ΤΜΗΜΑΤΟΣ Α ΑΝΑ ΧΙΛΙΟΜΕΤΡΙΚΗ ΘΕΣΗ.....	32
ΣΧΗΜΑ 4.4: ΠΑΧΗ ΤΜΗΜΑΤΟΣ Β ΑΝΑ ΧΙΛΙΟΜΕΤΡΙΚΗ ΘΕΣΗ.....	32
ΣΧΗΜΑ 4.5: ΔΙΑΦΟΡΑ ΣΦΑΛΜΑΤΩΝ TRAIN ΚΑΙ DEVELOPMENT	46
ΣΧΗΜΑ 5.1: ΠΡΟΒΛΕΨΗ ΤΩΝ ΔΕΙΚΤΩΝ IRI ΤΗΣ ΑΡΙΣΤΕΡΗΣ ΛΩΡΙΔΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ Α ΚΑΙ ΣΥΓΚΡΙΣΗ ΜΕ ΤΟΥΣ ΠΡΑΓΜΑΤΙΚΟΥΣ ΔΕΙΚΤΕΣ ΓΙΑ ΤΟ ΙΔΙΟ ΕΤΟΣ – 1 ^Η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ	57
ΣΧΗΜΑ 5.2: ΠΡΟΒΛΕΨΗ ΤΩΝ ΔΕΙΚΤΩΝ IRI ΤΗΣ ΔΕΞΙΑΣ ΛΩΡΙΔΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ Α ΚΑΙ ΣΥΓΚΡΙΣΗ ΜΕ ΤΟΥΣ ΠΡΑΓΜΑΤΙΚΟΥΣ ΔΕΙΚΤΕΣ ΓΙΑ ΤΟ ΙΔΙΟ ΕΤΟΣ – 2 ^Η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ	58
ΣΧΗΜΑ 5.3: ΠΡΟΒΛΕΨΗ ΤΩΝ ΔΕΙΚΤΩΝ IRI ΤΗΣ ΑΡΙΣΤΕΡΗΣ ΛΩΡΙΔΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ Β ΚΑΙ ΣΥΓΚΡΙΣΗ ΜΕ ΤΟΥΣ ΠΡΑΓΜΑΤΙΚΟΥΣ ΔΕΙΚΤΕΣ ΓΙΑ ΤΟ ΙΔΙΟ ΕΤΟΣ – 3 ^Η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ	60
ΣΧΗΜΑ 5.4: ΠΡΟΒΛΕΨΗ ΤΩΝ ΔΕΙΚΤΩΝ IRI ΤΗΣ ΔΕΞΙΑΣ ΛΩΡΙΔΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ Β ΚΑΙ ΣΥΓΚΡΙΣΗ ΜΕ ΤΟΥΣ ΠΡΑΓΜΑΤΙΚΟΥΣ ΔΕΙΚΤΕΣ ΓΙΑ ΤΟ ΙΔΙΟ ΕΤΟΣ – 4 ^Η ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ	61
ΣΧΗΜΑ 5.5: ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ 1 ^{ΗΣ} ΚΑΙ 2 ^{ΗΣ} ΠΕΙΡΑΜΑΤΙΚΗΣ ΑΝΑΛΥΣΗΣ	62
ΣΧΗΜΑ 5.6: ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ 3 ^{ΗΣ} ΚΑΙ 4 ^{ΗΣ} ΠΕΙΡΑΜΑΤΙΚΗΣ ΑΝΑΛΥΣΗΣ	63

1. ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο

Η αξιολόγηση και συντήρηση των οδοστρωμάτων μίας οδού ή ενός οδικού δικτύου, είναι ένα σημαντικό αντικείμενο πολιτικού μηχανικού, καθώς αποτελεί καθοριστικό παράγοντα στην άνεση και στην ασφάλεια των χρηστών. Ο χρόνος και το κόστος επέμβασης για τη συντήρηση ενός οδοστρώματος, μπορεί να βελτιστοποιηθούν όταν η επέμβαση πραγματοποιηθεί έγκαιρα. Η πρόβλεψη είναι η καλύτερη μέθοδος για την κατάλληλη προετοιμασία των μέσων και την άμεση δράση στο οδόστρωμα.

Στο πλαίσιο διαχείρισης και συντήρησης των οδοστρωμάτων, απαραίτητη είναι η εκτίμηση και αξιολόγηση διαφόρων παραμέτρων. Μία από αυτές τις παραμέτρους είναι η ομαλότητα του οδοστρώματος. Τα στοιχεία κυκλοφορίας, οι καιρικές και περιβαλλοντικές συνθήκες, καθώς και η δομική κατάσταση του οδοστρώματος παίζουν καθοριστικό ρόλο στη μείωση της ομαλότητας. Για την αξιολόγηση της ομαλότητας, χρησιμοποιούνται διάφοροι δείκτες, με πιο γνωστό τον πλέον ευρέως διαδεδομένο δείκτη IRI (International Roughness Index), ενώ έχουν γίνει πολλές έρευνες για την ανάπτυξη μοντέλων πρόβλεψης της ομαλότητας, με στόχο τον έγκυρο προγραμματισμό εργασιών συντήρησης του οδοστρώματος.

Από την άλλη μεριά, τα Νευρωνικά Δίκτυα καλούνται να φέρουν λύση σε πολύπλοκα προβλήματα, αγνοώντας τους θεμελιώδεις νόμους της φυσικής προκειμένου να αποδίδουν καλά στις παρατηρήσεις. Ένα από αυτά τα προβλήματα είναι και η πρόβλεψη της ομαλότητας των οδοστρωμάτων.

Τα Νευρωνικά Δίκτυα αποτελούν ένα είδος μηχανικής μάθησης. Η κατασκευή τους βασίζεται στην κατανόηση της λειτουργίας του ανθρώπινου εγκεφάλου, αποτελούμενου από νευρώνες, από όπου και πήραν το όνομά τους (Mazari & D.Rodriguez, 2016). Όπως έχει αποδειχθεί, ο ανθρώπινος εγκέφαλος μαθαίνει από εμπειρίες. Έτσι τα Νευρωνικά Δίκτυα, αποδίδοντας καλά στις παρατηρήσεις, δίνουν λύσεις σε προβλήματα που δεν μπορούν να γίνουν αντιληπτά από τον άνθρωπο ή να αντιμετωπιστούν με άλλες υπολογιστικές τεχνικές (Maind & Wankar, 2014).

Τις τελευταίες δεκαετίες, έχουν διεξαχθεί πολλές έρευνες για την πρόβλεψη της ομαλότητας των οδοστρωμάτων, με τη χρήση Νευρωνικών Δικτύων, (Mazari &

D.Rodriguez, 2016) (Georgiou, Plati, & Loizos, 2018) (El-Hakim & El-Badawy, 2013) (Georgiou, Plati, & Loizos, 2015). Η ακρίβεια όμως των αποτελεσμάτων πρόβλεψης και οι συνθήκες εφαρμογής των σχετικών μοντέλων για την πρόβλεψη της ομαλότητας των οδοστρωμάτων συνεχίζουν να αποτελούν ένα καίριο αντικείμενο έρευνας.

1.2 Στόχος και μεθοδολογία

Με δεδομένες τις ανάγκες βελτιστοποίησης και έγκαιρης συντήρησης των οδοστρωμάτων, στόχος της παρούσας διπλωματικής εργασίας είναι η ανάδειξη των Νευρωνικών Δικτύων ως ένα εργαλείο πρόβλεψης της ομαλότητας του οδοστρώματος μίας οδού.

Σε πρώτη φάση, πραγματοποιείται βιβλιογραφική έρευνα αναφορικά με την ομαλότητα, τους δείκτες και τους παράγοντες επιρροής της. Επίσης, πραγματοποιείται βιβλιογραφική έρευνα αναφορικά με τα είδη, τη δομή και τους τρόπους εκπαίδευσης των Νευρωνικών Δικτύων.

Σε δεύτερη φάση, αξιοποιούνται στοιχεία από τη βάση δεδομένων του Εργαστηρίου Οδοστρωμάτων του Εθνικού Μετσόβιου Πολυτεχνείου, προερχόμενα από δύο τμήματα ενός αυτοκινητοδρόμου. Τα στοιχεία αυτά αφορούν τους δείκτες ομαλότητας IRI, τα πάχη του οδοστρώματος και τα στοιχεία κυκλοφορίας, για εννέα συναπτά έτη (2013-2021).

Εν συνεχεία, πραγματοποιούνται τέσσερις πειραματικές αναλύσεις μέσω Νευρωνικών Δικτύων. Σε κάθε μία από αυτές, το δίκτυο εκπαιδεύεται με τα στοιχεία συλλογής του πρώτου τμήματος του αυτοκινητοδρόμου. Στις πρώτες δύο πειραματικές αναλύσεις, προβλέπονται οι δείκτες ομαλότητας IRI για το πρώτο τμήμα του αυτοκινητοδρόμου, για το έτος 2021, ενώ στις επόμενες δύο, προβλέπονται οι δείκτες ομαλότητας IRI για το δεύτερο τμήμα της οδού, για το ίδιο έτος. Μέσω της σύγκρισης των αποτελεσμάτων των αναλύσεων και των πραγματικών δεικτών IRI από τα στοιχεία της βάσης δεδομένων, διερευνάται η ικανότητα πρόβλεψης της ομαλότητας του οδοστρώματος ενός τμήματος μίας οδού, μέσω των Νευρωνικών Δικτύων, αλλά και η γενίκευση του αναπτυσσόμενου μοντέλου, για την πρόβλεψη της ομαλότητας των οδοστρωμάτων σε τμήματα άλλων οδών.

Επιπροσθέτως, σε όλες τις πειραματικές αναλύσεις χρησιμοποιούνται πέντε διαφορετικές μέθοδοι προβλέψεων, εκ των οποίων οι τρεις μέθοδοι αποτελούν γραμμικές συναρτήσεις, ενώ οι άλλες δύο αποτελούν Νευρωνικά Δίκτυα. Έτσι, με τη σύγκριση των αποτελεσμάτων, αναδεικνύεται η καλύτερη μέθοδος πρόβλεψης της ομαλότητας.

1.3 Δομή εργασίας

Η παρούσα διπλωματική εργασία, συμπεριλαμβανομένου του παρόντος, αποτελείται από 7 κεφάλαια, τα οποία περιγράφονται αναλυτικά ακολούθως.

Στο δεύτερο κεφάλαιο, αναφέρονται τα γενικά στοιχεία της ομαλότητας και οι παράγοντες επιρροής αυτής. Έμφαση δίνεται στον διεθνή δείκτη ομαλότητας IRI, ο οποίος χρησιμοποιείται στο πλαίσιο της παρούσας εργασίας.

Στο τρίτο κεφάλαιο, γίνεται μία εισαγωγή στα Νευρωνικά Δίκτυα και αναλύονται τα είδη των αρχιτεκτονικών, καθώς και οι τεχνικές εκπαίδευσής τους.

Στο τέταρτο κεφάλαιο, περιγράφεται η αναλυτική διαδικασία που ακολουθείται. Ξεκινώντας, δίνεται μια συνοπτική περιγραφή της διαδικασίας και παρουσιάζεται η ανάλυση των δεδομένων. Ακολουθεί η περιγραφή των μοντέλων που χρησιμοποιήθηκαν για την πρόβλεψη των δεικτών IRI, αναλύοντας τη διαδικασία εισαγωγής των δεδομένων, καθώς και τη διαδικασία εκπαίδευσης των Νευρωνικών Δικτύων.

Στο πέμπτο κεφάλαιο, περιλαμβάνονται τα αποτελέσματα της βέλτιστης μεθόδου, τα σφάλματα που προκύπτουν από την παραπάνω αναλυτική διαδικασία και γίνεται συγκριτική ανάλυση των σφαλμάτων.

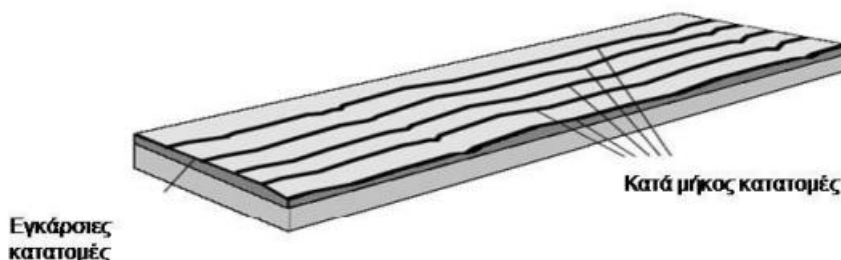
Στο έκτο κεφάλαιο, παραθέτονται τα συμπεράσματα που προέκυψαν από την αξιολόγηση των αποτελεσμάτων.

Στο έβδομο και τελευταίο κεφάλαιο, αναφέρεται όλη η βιβλιογραφία που χρησιμοποιήθηκε για την εκπόνηση της παρούσας διπλωματικής εργασίας.

2. ΟΜΑΛΟΤΗΤΑ ΟΔΟΣΤΡΩΜΑΤΟΣ

2.1 Ορισμός και παράγοντες επιρροής

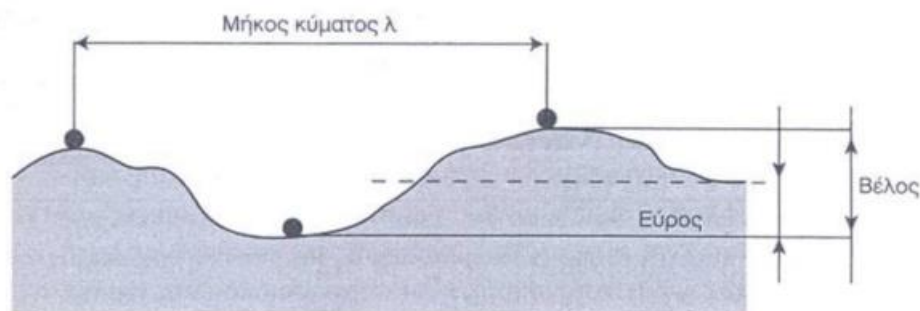
Η ομαλότητα είναι μία από τις βασικές παραμέτρους που συμβάλλουν στην αξιολόγηση της λειτουργικής κατάστασης του οδοστρώματος, καθώς σύμφωνα με αυτή προσδιορίζονται οι αποκλίσεις της επιφάνειας του οδοστρώματος από την επιπεδότητα, είτε κατά μήκος είτε εγκάρσια αυτής (Λοΐζος & Πλατή, 2021). Η έννοια «προφίλ» της επιφάνειας του οδοστρώματος κατά τις δύο αυτές κατευθύνσεις απεικονίζεται στην Εικόνα 2.1.



Εικόνα 2.1: Επιφάνεια οδοστρώματος και κατανομές (προφίλ)

Πηγή: (Λοΐζος & Πλατή, 2021)

Οι κυματισμοί είναι το φυσικό μέγεθος με το οποίο εκφράζεται η ομαλότητα (Λοΐζος & Πλατή, 2021)(Εικόνα 2.2). Οι κυματισμοί μικρού μήκους ($< 3 \text{ m}$) είναι συνήθως αποτέλεσμα των επιφανειακών ατελειών του οδοστρώματος, ενώ οι κυματισμοί μεγάλου μήκους ($> 10 \text{ m}$) οφείλονται κυρίως στις υποκείμενες στρώσεις. Η εμφάνιση αυτών των παραμορφώσεων τείνει να αυξάνεται με την αύξηση των αξονικών διελεύσεων, δηλαδή με την αύξηση του κυκλοφοριακού φόρτου. Εξίσου σημαντικός παράγοντας, που συμβάλλει στην αύξηση των κατά μήκος κυματισμών, είναι και ο τύπος των διερχόμενων οχημάτων αλλά και ο τρόπος κατασκευής της οδού.



Εικόνα 2.2: Στοιχεία αστοχιών επιτεδότητας
 Πηγή: (Λοΐζος & Πλατή, 2021)

Το φαινόμενο της έλλειψης ομαλότητας συμβάλλει αρνητικά στα λειτουργικά χαρακτηριστικά του οδοστρώματος αλλά και στη φθορά των τροχών των οχημάτων, με αποτέλεσμα την αύξηση του κινδύνου πρόκλησης ενός ατυχήματος. Επιπλέον, με την αύξηση της απώλειας τριβής στο οδόστρωμα συνεπάγεται την ανάγκη μείωσης της ταχύτητας του οχήματος, αλλά και πιθανές φθορές των τροχών. Επίσης, λόγω της έλλειψης ομαλότητας μπορεί να προκληθεί αναπήδηση των τροχών του οχήματος με αποτέλεσμα τη φθορά της ανάρτησης, των ελαστικών αλλά και του οδοστρώματος. Τέλος, η πρόκληση αναταράξεων στο όχημα αλλά και η απώλεια ελέγχου του οχήματος λόγω της ποιότητας κύλισης, επηρεάζουν την ασφάλεια των επιβατών.

Οι βασικοί παράγοντες που επιφέρουν τη μείωση της ποιότητας κύλισης, αλλά και τη φέρουσα ικανότητα του οδοστρώματος, είναι η αύξηση της κυκλοφορίας, οι τύποι των οχημάτων, οι καιρικές και περιβαλλοντικές συνθήκες και η γήρανση των υλικών. Πιο συγκεκριμένα, με την αύξηση της κυκλοφορίας, καθώς και με την αύξηση του επιβαλλόμενου κινητού φορτίου στην οδό, δημιουργούνται κυματισμοί στην επιφάνεια του οδοστρώματος, όπως αναφέρθηκε και στην παράγραφο 2.1 του κεφαλαίου. Αυτό έχει σαν αποτέλεσμα την αναπήδηση των τροχών και επομένως την ενόχληση των επιβατών. Ομοίως, οι αυξομειώσεις της θερμοκρασίας επιδρούν στα μηχανικά χαρακτηριστικά των υλικών ασφατικής στρώσεως και επομένως στην έλλειψη ομαλότητας. Εξίσου σημαντικό ρόλο έχουν και ο τρόπος κατασκευής της οδού, τα υλικά κατασκευής και τα πάχη των ασφατικών στρώσεων. Τέλος, όσο μεγαλύτερη είναι η γήρανση του οδοστρώματος, τόσο μικρότερη η φέρουσα ικανότητα και άρα υποβάθμιση της δομικής κατάστασής του.

2.2 Δείκτες έκφρασης ομαλότητας

Για την έκφραση της ομαλότητας χρησιμοποιούνται διάφοροι δείκτες, οι οποίοι προκύπτουν, έπειτα από την ανάλυση και επεξεργασία σχετικών μετρήσεων. Οι τρεις πιο διαδεδομένοι δείκτες είναι οι εξής:

- International Roughness Index (IRI)
- Ride Number (RN)
- Profile Index (PI)

Ο δείκτης «**International Roughness Index**», γνωστός για εν συντομία ως **IRI**, είναι ο πιο ευρέως διαδεδομένος δείκτης ομαλότητας. Η έννοια του δείκτη αναπτύχθηκε έπειτα από το πείραμα «International Road Roughness» το οποίο πραγματοποιήθηκε το 1982 στη Βραζιλία (Perera, Byrum, & Kohn, 1998). Ο δείκτης IRI αντιπροσωπεύει τους κραδασμούς που προκαλούνται σε ένα όχημα, εξαιτίας της ποιότητας κύλισης του οδοστρώματος.

Ο δείκτης «**Ride Number**», γνωστός για εν συντομία ως **RN**, αναπτύχθηκε στις αρχές του 1980 από τον Michael Janoff μέσα από μία έρευνα του National Cooperative Highway Research Program (Sayers & Karamihis, 1996). Η διαδικασία υπολογισμού πραγματοποιήθηκε με τη βοήθεια των προφιλόμετρων, που είχαν μόλις εφευρεθεί από τους Elson Spangler και William Kelly, για τη δημιουργία μιας σχέσης που να συνδέει τη μέτρηση προφίλ ενός οδοστρώματος με την ποιότητα κύλισης (Spangler & Kelly, 1994).

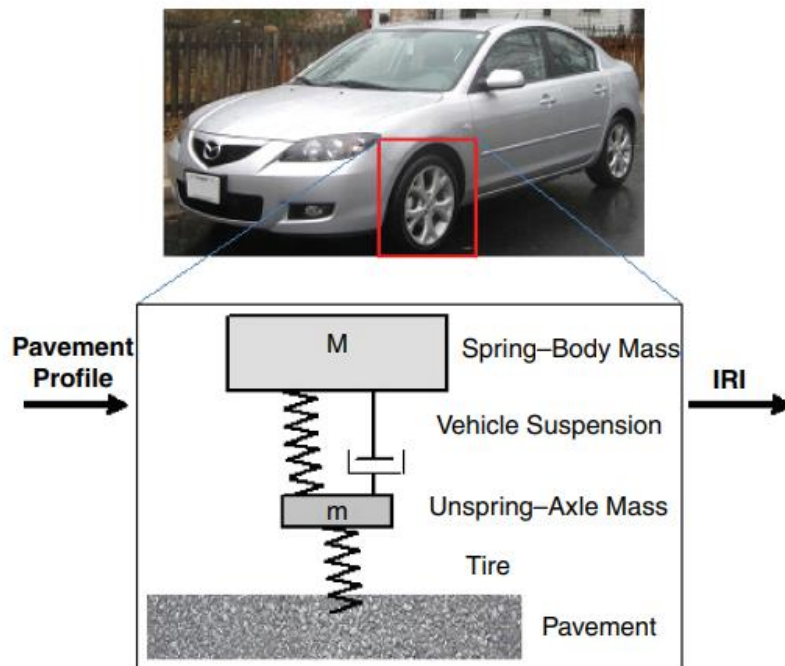
Ο δείκτης «**Profile Index**», γνωστός και ως **PI**, προέκυψε επίσης από μετρήσεις που έγιναν με τη βοήθεια προφίλογράφου, και χρησιμοποιείται για την αποδοχή ή όχι της ομαλότητας κυρίως νέων οδοστρωμάτων (Mondal, Hand, & Ward, 2000).

2.3 Διεθνής δείκτης ομαλότητας (IRI)

2.3.1 Γενικά

Ο διεθνής δείκτης ομαλότητας IRI, προέκυψε έπειτα από πείραμα το οποίο πραγματοποιήθηκε το 1982 στη Βραζιλία από την Παγκόσμια Τράπεζα (World Bank) (Perera, Byrum, & Kohn, 1998).

Για τον υπολογισμό του δείκτη IRI χρησιμοποιήθηκε το προσομοίωμα του «ενός τετάρτου του οχήματος» ενός μόνο τροχού. Το προσομοίωμα αυτό αποτελείται από ένα ελαστικό με κατακόρυφο ελατήριο, τη μάζα του άξονα που στηρίζεται από το ελαστικό, ένα ελατήριο ανάρτησης, την απόσβεση και τη μάζα του οχήματος που υποστηρίζεται από την ανάρτηση (Islam, Buttlar, Aldunate, & Vavrik, 2015)(Εικόνα 2.3).



Εικόνα 2.3: Προσομοίωμα ενός τετάρτου του οχήματος
Πηγή: (Islam, Buttlar, Aldunate, & Vavrik, 2015)

Στο όχημα του ενός τετάρτου, η ταχύτητα προσομοίωσης κίνησης είναι 80 km/h. Η προσομοιωμένη κίνηση ανάρτησης διαιρείται με το μήκος του προφίλ, δίνοντας τον δείκτη IRI (ΜΥΨΚΑ, 2012). Ο δείκτης IRI μετριέται σε μονάδες κλίσεις m/km.

Οι τιμές του δείκτη IRI, γενικά, κυμαίνονται μεταξύ 0 και 20. Πρακτικά όμως, για τιμές του δείκτη μεγαλύτερες από 8, το οδόστρωμα είναι αδιάβατο.

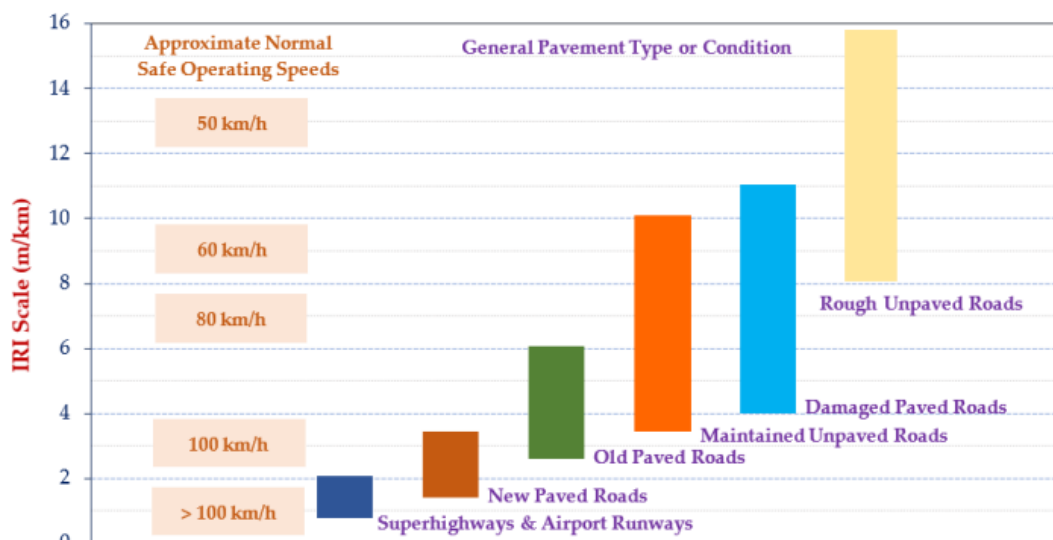
2.3.2 Πλεονεκτήματα

Σε σχέση με τους άλλους δείκτες, ο IRI έχει τα εξής πλεονεκτήματα (Λοΐζος & Πλατή, 2021):

- Είναι ο πλέον διαδεδομένος δείκτης ομαλότητας, ο οποίος μπορεί να χρησιμοποιηθεί ευρέως από διάφορους τύπους συστημάτων καταγραφής της ομαλότητας
- Έχει συσχέτιση με την ταλάντωση των οχημάτων και επομένως και με τη συμπεριφορά τους
- Έχει αντικειμενική μέτρηση της ομαλότητας, χωρίς να παρεμποδίζεται από ανθρώπινους παράγοντες
- Η κλίμακά του έχει συσχέτιση με την κατανομή της επιφάνειας του οδοστρώματος και παραμένει σταθερή με το χρόνο
- Έχει μονάδες κλίσεις (m/km, in/km κλπ)

2.3.3 Όρια προδιαγραφών

Η τιμή του IRI ίση με μηδέν προσδιορίζει ένα απολύτως ομαλό οδόστρωμα. Επομένως, όσο πιο μικρός ο δείκτης, τόσο καλύτερη η ομαλότητα. Πιο συγκεκριμένα, η διαβάθμιση των δεικτών IRI για διάφορους τύπους οδοστρωμάτων, με διαφορετικές ταχύτητες κίνησης, απεικονίζεται πιο αναλυτικά στο Σχήμα 2.2. Επίσης, ο Πίνακας 2.1 περιλαμβάνει κάποιες ενδεικτικές προδιαγραφές ομαλότητας από τη διεθνή βιβλιογραφία.



Σχήμα 2.1: Διαβάθμιση δεικτών IRI για οδοστρώματα με διαφορετικά όρια ταχυτήτων

Πηγή: (Chen, Lin, Tang, Chu, & Cheng, 2020)

Πίνακας 2.1: Επιθυμητά όρια δεικτών ομαλότητας IRI σε οδοστρώματα με διαφορετικά όρια ταχυτήτων

Επίπεδο ποιότητας	Όρια δεικτών IRI σε διαφορετικές ταχύτητες (m/km)					
	20	40	60	80	100	120
Πολύ καλό	< 5,72	< 2,86	< 1,90	< 1,43	< 1,14	< 0,95
Καλό	5,72 - 8,99	2,86 - 4,49	1,90 - 2,99	1,43 - 2,24	1,14 - 1,79	0,95 - 1,49
Σχεδόν καλό	9,00 - 11,39	4,50 - 5,69	3,00 - 3,79	2,25 - 2,84	1,80 - 2,27	1,50 - 1,89
Μέτριο	11,40 - 16,16	5,70 - 8,08	3,80 - 5,40	2,85 - 4,05	2,28 - 3,24	1,90 - 2,70
Ανεπαρκές	> 16,16	> 8,08	> 5,40	> 4,05	> 3,24	> 2,70

Πηγή: (Chen, Lin, Tang, Chu, & Cheng, 2020)

Σημειώνεται ότι οι τιμές του Πίνακα 2.1 είναι ένα παράδειγμα. Οι αποδεκτές τιμές του δείκτη IRI σε ένα οδόστρωμα σχετίζονται πάντα με τις προδιαγραφές του έργου.

2.4 Συστήματα καταγραφής

Για την καταγραφή της ομαλότητας χρησιμοποιούνται εξειδικευμένα συστήματα καταγραφής επιφανειακών χαρακτηριστικών. Οι πέντε βασικές κατηγορίες εξοπλισμού που χρησιμοποιούνται διεθνώς για την καταγραφή της ομαλότητας είναι οι εξής:

- Συστήματα καταγραφής μηχανικής απόκρισης (Response Type Road Roughness Measuring Systems RTRRMS)
- Υψηλής ταχύτητας προφιλόμετρα (High speed inertial profilers)
- Προφιλογράφοι (Profilographs)
- «Ελαφρά» προφιλόμετρα (Lightweight profilers)
- Χειροκίνητες συσκευές (Manual devices)

Τα **συστήματα καταγραφής μηχανικής απόκρισης (Response Type Road Roughness Measuring Systems RTRRMS)** αποτελούνται από ένα αυτοκινούμενο όχημα ή ένα ρυμουλκούμενο με ένα ή δύο τροχούς και επιπλέον ένα οδόμετρο ενσωματωμένο για τη μέτρηση των παραμορφώσεων της ανάρτησης. Η μέτρηση του δείκτη ομαλότητας προκύπτει στα αυτοκινούμενα οχήματα από τη συνάθροιση της κατακόρυφης κίνησης των πίσω τροχών σε σχέση με το σκελετό του οχήματος, ενώ

στα ρυμουλκούμενα προκύπτει από τη συνάθροιση από την κατακόρυφη κίνηση του τρέιλερ σε σχέση με το σκελετό του οχήματος (Λοΐζος & Πλατή, 2021).

Τα **υψηλής ταχύτητας προφιλόμετρα (High speed inertial profilers)** είναι τα πιο ευρέως διαδεδομένα συστήματα μέτρησης της ομαλότητας του οδοστρώματος. Αποτελούνται από ένα φορτηγό τύπου βαν, εξοπλισμένο με ένα επιταχυνσιόμετρο και ένα ή περισσότερους αισθητήρες λέιζερ. Οι αισθητήρες λέιζερ είναι τοποθετημένοι στο μπροστινό μέρος (προφυλακτήρα) του βαν και μετρούν την κατακόρυφη απόσταση από το αμάξωμα του οχήματος μέχρι την επιφάνεια του οδοστρώματος (Nazef, Mraz, Iyer, & Choubane, 2009). Το επιταχυνσιόμετρο αντισταθμίζει την κατακόρυφη κίνηση του αμαξώματος του οχήματος.

Οι **προφίλογράφοι (Profilographs)** χρησιμοποιήθηκαν για πρώτη φορά στις αρχές του 1940 για τη μέτρηση προφίλ οδοστρωμάτων από σκυρόδεμα. Στη συνέχεια, χρησιμοποιήθηκαν για τον έλεγχο της ομαλότητας των ασφαλικών οδοστρωμάτων. Αποτελούνται από μία δοκό υποστηριζόμενη από τροχούς, η οποία σύρεται ή ρυμουλκείται με μικρές ταχύτητες κατά μήκος της οδού (Devore, Hossain, & William H. Parcells, 1995).

Τα **“Ελαφρά” προφιλόμετρα (Lightweight profilers)** είναι πολύ μικρότερα και ελαφρύτερα από τα προφιλόμετρα υψηλών ταχυτήτων. Παρόλο που η αρχή λειτουργίας τους σε σχέση με τα προφιλόμετρα υψηλών ταχυτήτων είναι ίδια, έχουν το πλεονέκτημα ότι, λόγω του μικρού βάρους τους, μπορούν να χρησιμοποιηθούν αμέσως μετά την κατασκευή της ασφάλτου θερμού μίγματος (Mondal, Hand, & Ward, 2000).

Οι **χειροκίνητες συσκευές (Manual devices)** χωρίζονται στην τοπογραφική αποτύπωση της επιφάνειας των οδοστρωμάτων, στη συσκευή Walking Profiler και στη συσκευή Dipstick. Η διαδικασία της τοπογραφικής αποτύπωσης της επιφάνειας του οδοστρώματος είναι η πιο ακριβής μέθοδος καταγραφής της γεωμετρικής κατανομής του οδοστρώματος, αλλά ταυτόχρονα και η πιο χρονοβόρα και για αυτό δεν προτιμάται (Yousefzadeh, Azadi, & Soltani, 2009).

3. ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

3.1 Εισαγωγή στα Νευρωνικά Δίκτυα

3.1.1 Γενικά

Η πρόβλεψη και η μοντελοποίηση της συμπεριφοράς μη γραμμικών και πολύπλοκων δυναμικών συστημάτων, αποτελεί πρόκληση για τον τομέα της επιστήμης μέχρι και σήμερα. Για την αντιμετώπιση αυτής της πρόκλησης, οι ερευνητές τείνουν να συλλέγουν όλο και περισσότερα στοιχεία παρατήρησης. Η συσχέτιση, η ανάλυση και η κατανόηση αυτών των στοιχείων, αποτελεί ένα ευρύ πεδίο έρευνας και καινοτομίας. Στο πεδίο αυτό, τις τελευταίες δύο δεκαετίες περίπου, έχει εισχωρήσει η μηχανική μάθηση για την επεξεργασία και ανάλυση δεδομένων μεγάλων όγκων. Με τη μηχανική μάθηση, αγνοώντας τους θεμελιώδεις νόμους της φυσικής και αποδίδοντας καλά στις παρατηρήσεις, έχει επιτευχθεί η συσχέτιση διαφόρων δεδομένων συλλογής, παρουσιάζοντας εύλογες προβλέψεις.

Ένα είδος μηχανικής μάθησης είναι τα Νευρωνικά Δίκτυα. Παραδοσιακά, ο όρος «Νευρωνικό Δίκτυο» αναφέρεται στους βιολογικούς νευρώνες, αλλά η σύγχρονη χρήση του όρου αναφέρεται στα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks - ANN) (Maind & Wankar, 2014). Τα ANN αποτελούνται από μαθηματικά μοντέλα εμπνευσμένα από τη δομή βιολογικών νευρικών συστημάτων (Mazari & D.Rodriguez, 2016). Οι αλγόριθμοι αυτοί χρησιμοποιούνται για την επίλυση πολύπλοκων προβλημάτων, όπως είναι τα μη γραμμικά προβλήματα. Τα ANN αποτελούνται από ομάδες υπολογιστικών στοιχείων, τους νευρώνες. Μέσα σε αυτά τα υπολογιστικά στοιχεία διέρχονται δεδομένα στα οποία εκτελούνται μαθηματικές πράξεις. Υπάρχουν διάφοροι τύποι τεχνητών νευρωνικών δικτύων, οι οποίοι διαφέρουν μεταξύ τους στην αρχιτεκτονική σχεδιάσή τους. Η επιλογή αρχιτεκτονικής σχεδίασης εξαρτάται από τη μορφή του προβλήματος που καλείται να επιλύσει το νευρωνικό δίκτυο.

Μέσω της διεθνούς βιβλιογραφίας εντοπίζονται διάφορες εφαρμογές των Νευρωνικών Δικτύων στους τομείς της ιατρικής, της πληροφορικής, της οικονομίας κ.ά. Κάποια παραδείγματα της χρήσης τους είναι η αναγνώριση ομιλίας, η πρόβλεψη του καιρού, αλλά και η διάγνωση ασθενειών με βάση τα συμπτώματα των ασθενών.

3.1.2 Ορισμός

Ένα νευρωνικό δίκτυο είναι μια αλληλουχία από νευρώνες, συναρτήσεις ενεργοποίησης, καθώς και άλλα στοιχεία, με σκοπό να μοιάσει σε λειτουργικότητα στο ζωικό νευρώνα, από τον οποίο εμπνέεται. Η ικανότητα επεξεργασίας των δεδομένων προκύπτει από μια διαδικασία προσαρμογής ή μάθησης, μέσω ενός συνόλου μοτίβων προπόνησης (Gurney, 1997). Όσο περισσότερο προπονείται το δίκτυο, τόσο περισσότερο καλά μαθαίνει από τα δεδομένα που του δίνονται. Αυτή η μάθηση, αντικατοπτρίζεται μαθηματικά με την εκμάθηση βαρών, δηλαδή αριθμητικών τιμών παραμέτρων, βάσει των οποίων υπολογίζεται στη συνέχεια η πρόβλεψη (αποτέλεσμα).

3.1.3 Ιστορική Αναδρομή

Το 1943, ο νευροφυσιολόγος Warren McCulloch και ο μαθηματικός Walter Pitts παρουσιάζουν για πρώτη φορά το μοντέλο του νευρωνικού δικτύου. Στην έρευνά τους με τίτλο «The Logical Calculus of the Ideas Immanent in Nervous Activity» παρουσιάζουν έναν νευρώνα ως ένα «διακόπτη» που λαμβάνει είσοδο από άλλους νευρώνες. Ο νευρώνας «διακόπτης» αυτός ενεργοποιείται ή απενεργοποιείται, ανάλογα με τη συνολική σταθμισμένη εισροή δεδομένων (Smith & Gupta, 2000).

Το 1944, οι Joseph Erlanger και Herbert Spencer Gasser εντόπισαν διάφορες ποικιλίες νευρικών ινών και ανέπτυξαν τη σχέση μεταξύ της διαμέτρου της ίνας και της ταχύτητας του δυναμικού δράσης (Macukow, 2016).

Το 1949, ο ψυχολόγος Donald Hebb επισήμανε στο βιβλίο του με τίτλο «The Organization of Behavior» ότι τα νευρωνικά δίκτυα ενισχύονται κάθε φορά που χρησιμοποιούνται, μια θεωρία θεμελιώδης για τη ανάπτυξη των νευρωνικών δικτύων (Smith & Gupta, 2000).

Ένα μεγάλο βήμα στην ιστορική και τεχνολογική εξέλιξη των νευρωνικών δικτύων έγινε όταν ξεκίνησαν να αναπτύσσονται τα πρώτα συστήματα μάθησης. Το πρώτο σύστημα μάθησης αναπτύχθηκε το 1958, όταν ο ψυχολόγος Frank Rosenblatt, μέσα από το βιβλίο του «Principles of Neurodynamics», διεξήγαγε ερευνητική μελέτη για τη συσκευή Perceptron. Η συσκευή αυτή είχε κατασκευαστεί σύμφωνα με τις βιολογικές αρχές και παρουσίαζε ικανότητες μάθησης. Στη συνέχεια, το 1960, αναπτύχθηκε το σύστημα ADELIN (ADaptive LINEar Element) από τον καθηγητή ηλεκτρολόγων μηχανικών του πανεπιστημίου του Στάνφορντ, Professor Bernard

Widrow, και το διδακτορικό φοιτητή του, Ted Hoff. Σε αυτό το σύστημα χρησιμοποιήθηκε διαφορετική μέθοδος μάθησης από ότι στο Perceptron, γνωστή ως Least-Mean-Squares. Έπειτα, από τους ίδιους ερευνητές, αναπτύχθηκε μια διαδικασία εκμάθησης, η οποία εξέταζε την τιμή πριν γίνει η προσαρμογή της στο βάρος (Marini, Bucci, Magrì, & Magrì, 2008).

Μια περίοδος διχασμού ξεκίνησε όταν οι Marvin Minsky και Seymour Papert, μέσα από το βιβλίο τους «Perceptrons: An Introduction to Computational Geometry», το 1969, παρουσίασαν μια σειρά από προβλήματα της συσκευής Perceptron, τονίζοντας ότι μπορεί να χειριστεί μόνο γραμμικά προβλήματα (Smith & Gupta, 2000).

Αυτό όμως δε σταμάτησε την αναζήτηση και το ενδιαφέρον κάποιων ερευνητών. Πιο συγκεκριμένα, ακολούθησαν σημαντικές εξελίξεις για την ανάπτυξη των νευρωνικών δικτύων. Αξιοσημείωτη είναι η βάση για μάθηση σε τεχνητούς νευρώνες βασιζόμενη στη βιολογική αρχή, από τον Klorf το 1972. Το 1974, ο Paul Werbos ανέπτυξε τη μέθοδο εκμάθησης αναδρομικής διάδοσης. Το 1975, ο επιστήμονας Kunihiko Fukushima ανέπτυξε ένα πολυεπίπεδο νευρωνικό δίκτυο για την ερμηνεία χειρόγραφων χαρακτήρων. Το 1976, ο Stephen Grossberg, εισήγαγε την έννοια του προσαρμοστικού συντονισμού, μια θεωρία της ανθρώπινης γνωστικής επεξεργασίας πληροφοριών (Macukow, 2016).

Ερχόμενοι στον 21^ο αιώνα, η πρόοδος που έχει σημειωθεί στον τομέα των νευρωνικών δικτύων είναι ραγδαία, με αποτέλεσμα τη μεγάλη προσέλκυση χρηματοδοτήσεων για περαιτέρω έρευνα.

Ξεκινώντας το 2009, μέχρι και το 2012, ο Juergen Schmidhuber και η ερευνητική ομάδα του, ανέπτυξαν τα νευρωνικά δίκτυα βαθιάς μάθησης (deep learning in neural networks). Τέλος το 2014, οι επιστήμονες της IBM, δημιούργησαν τον επεξεργαστή TrueNorth, του οποίου η αρχιτεκτονική μοιάζει με αυτή του ανθρώπινου εγκεφάλου. Η IBM ανέπτυξε ένα ολοκληρωμένο κύκλωμα που έχει τη δυνατότητα προσομοίωσης των εκατομμυρίων νευρώνων και 256 εκατομμυρίων συνάψεων σε πραγματικό χρόνο. Το σύστημα αυτό μπορεί να εκτελεί από 46 έως και 400 δισεκατομμύρια συναπτικές λειτουργίες κάθε δευτερόλεπτο (Macukow, 2016).

3.1.4 Πλεονεκτήματα Νευρωνικών Δικτύων

Κάποια από τα πλεονεκτήματα που έχουν τα νευρωνικά δίκτυα, συγκριτικά με άλλες μεθόδους επίλυσης κάποιου προβλήματος, είναι τα εξής:

- Δεν απαιτείται κάποιος αλγόριθμος βασισμένος στη φυσική για την κατασκευή του μοντέλου. Το νευρωνικό δίκτυο, λαμβάνοντας όλα τα δεδομένα εισαγωγής, αναπτύσσει μια μαθηματική σχέση και προπονείται ώστε να δώσει μια μελλοντική εκτίμηση.
- Η μοντελοποίηση είναι ταχύτερη και πιο ευέλικτη από κάθε άλλη προσέγγιση βασιζόμενη στη φυσική, όταν πρόκειται για την επίλυση ενός πολύ περίπλοκου μαθηματικού προβλήματος (Yadav, Yadav, & Kumar, 2015).
- Μπορούν να χειριστούν και μη γραμμικές σχέσεις.
- Οι χρήστες μπορούν να ενσωματώσουν την τεχνογνωσία, καθώς και την εμπειρία τους.
- Τα νευρωνικά δίκτυα μπορούν να αντιμετωπίσουν δύσκολα προβλήματα που προκύπτουν σε πολλές επιστήμες και εφαρμογές μηχανικής, σε πραγματικό χρόνο (Yadav, Yadav, & Kumar, 2015).

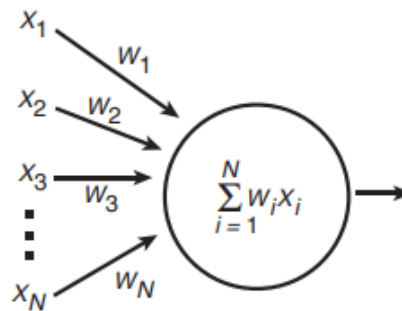
3.2 Δομή του νευρώνα

Η δομή του τεχνητού νευρώνα αναπτύχθηκε με βάση τα τεχνητά μοντέλα του ανθρώπινου νευρικού συστήματος και εγκεφάλου. Πιο συγκεκριμένα, τα υπολογιστικά στοιχεία, ή οι τεχνητές μονάδες επεξεργασίας του νευρωνικού δικτύου, δηλαδή οι τεχνητοί νευρώνες, είναι απλουστευμένα μοντέλα βιολογικών νευρώνων.

Ο πιο βασικός ρόλος των τεχνητών νευρώνων είναι η μη γραμμική παροχή συνεχών εξόδων και η εκτέλεση απλών λειτουργιών. Ένα παράδειγμα αυτού, είναι η συλλογή σημάτων διαθέσιμων στις εισόδους τους, διατεταγμένων ανάλογα με τη λειτουργικότητά τους, παράγοντας μια έξοδο, έχοντας λάβει υπόψη τις απαιτούμενες λειτουργίες ενεργοποίησής τους.

Ο νευρώνας αποτελεί τη βασική μονάδα επεξεργασίας, που συλλέγει τα σήματα εισόδου, τα οποία μετασχηματίζει, και δίνει μια έξοδο στο δίκτυο ή σε άλλους νευρώνες. Οι είσοδοι που προέρχονται από το εξωτερικό περιβάλλον και

συγκεντρώνονται από τους δενδρίτες στο βιολογικό νευρώνα (Εικόνα 3.1), αποτελούνται από το σύνολο $\{x_1, x_2, x_3, \dots, x_n\}$. Η ζύγιση πραγματοποιείται στον τεχνητό νευρώνα ως σύνολο συνοπτικών βαρών $\{w_1, w_2, \dots, w_n\}$, μέσω των συναπτικών διασταυρώσεων. Το βάρος αυτό αντιπροσωπεύει την τιμή επίδρασης που έχει ο προηγούμενος νευρώνας.



Εικόνα 3.1: Δομή νευρώνα
Πηγή: (Krogh, 2008)

Η τροποποίηση της εισόδου επιτυγχάνεται χρησιμοποιώντας μια μη γραμμική συνάρτηση, γνωστή και ως συνάρτηση ενεργοποίησης. Ο υπολογισμός της συνάφειας καθεμίας από τις εισόδους των νευρώνων $\{x_i\}$, πραγματοποιείται έπειτα από των πολλαπλασιασμό αυτών με το συναπτικό τους βάρος $\{w_i\}$. Η συνολική είσοδος του τεχνητού κυψελωτού σώματος είναι το σταθμισμένο άθροισμα των εισόδων του (Gurney, 1997).

3.3 Αρχιτεκτονική Νευρωνικών Δικτύων

3.3.1 Βασικές κατηγορίες

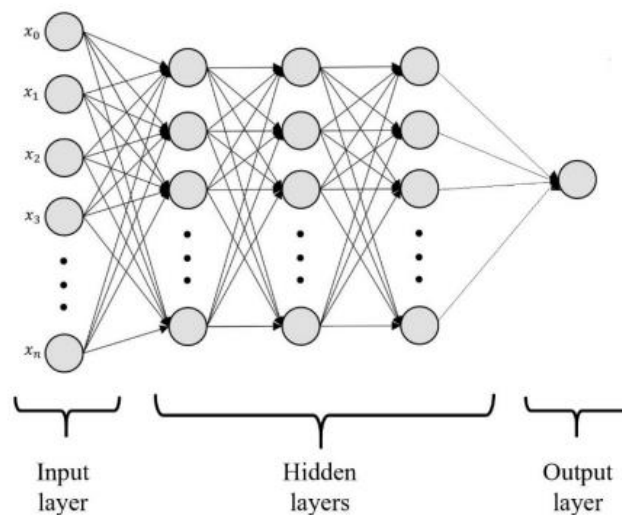
Η κατάλληλη αρχιτεκτονική σε ένα νευρωνικό δίκτυο, επιλέγεται με βάση το σκοπό κατασκευής του δικτύου. Οι τέσσερις βασικότερες κατηγορίες αρχιτεκτονικών νευρωνικών δικτύων είναι οι εξής:

1. Multilayer Perceptron (MLP)
2. Recurrent Neural Network (RNN)

3. Long Short Term Memory (LSTM)
4. Convolutional Neural Network (CNN)

3.3.2 Multilayer Perceptron (MLP)

Τα Multilayer Perceptrons (MLPs), γνωστά και ως γραμμικά δίκτυα, είναι μία από τις κοινώς χρησιμοποιούμενες μορφές αρχιτεκτονικής των ANN για μελέτες πρόβλεψης. Τα δίκτυα αυτά μπορούν να προσεγγίσουν οποιαδήποτε συνεχή συνάρτηση αλλά και μη γραμμικά προβλήματα. Κύριες χρήσεις των δικτύων αυτών είναι η ταξινόμηση προτύπων, η πρόβλεψη, η αναγνώριση κειμένων και η προσέγγιση. Τα MLP αποτελούνται από τρία επίπεδα, το επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα και το επίπεδο εξόδου (Εικόνα 3.2) (Domashova, Emtseva, Fail, & Gridin, 2021).



Εικόνα 3.2: Αρχιτεκτονική Multilayer perceptron (MLP)
Πηγή: (Domashova, Emtseva, Fail, & Gridin, 2021)

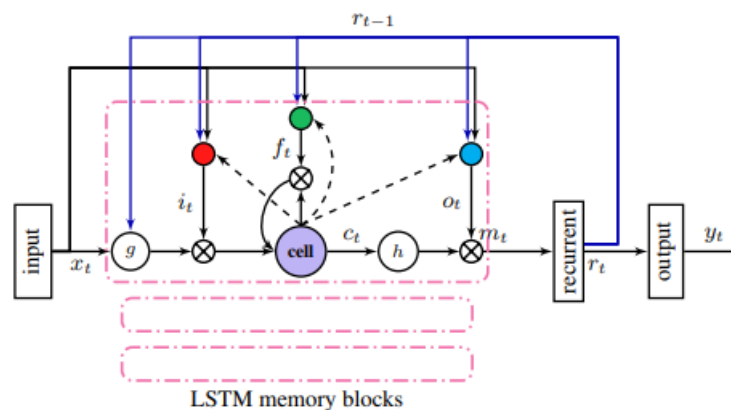
Το επίπεδο εισόδου λαμβάνει το σήμα εισόδου ως προς επεξεργασία. Στα κρυφά επίπεδα οι νευρώνες εκπαιδεύονται με τον αλγόριθμο εκμάθησης, όπως αναφέρθηκε στο υποκεφάλαιο 3.2. Στη συνέχεια τα δεδομένα ρέουν προς το επίπεδο εξόδου, όπου εκτελείται και η απαιτούμενη εργασία.

3.3.3 Recurrent Neural Network (RNN)

Μια άλλη κατηγορία αρχιτεκτονικής νευρωνικών δικτύων είναι τα Recurrent Neural Networks (RNNs). Τα RNN χρησιμοποιούνται σε εφαρμογές όπως η αναγνώριση ομιλίας, η αυτόματη μετάφραση, η μελλοντική πρόβλεψη και η ταξινόμηση. Παρόλο που τα RNN αποτελούν ένα ισχυρό μοντέλο κωδικοποίησης, συχνά παρουσιάζουν προβλήματα απώλειας μακροχρόνιας μνήμης, μέσω του προβλήματος του «vanishing gradient» (Pienaar & Malekian, 2019). Το φαινόμενο αυτό παρατηρείται κατά τη συσσώρευση μεγάλων διαβαθμίσεων σφάλματος, που έχει ως συνέπεια τη σημαντικές αλλαγές στο μοντέλο του νευρωνικού δικτύου, κατά τη διάρκεια της εκπαίδευσης, με αποτέλεσμα την αστάθεια του εκπαιδευόμενου μοντέλου.

3.3.4 Long Short Term Memory (LSTM)

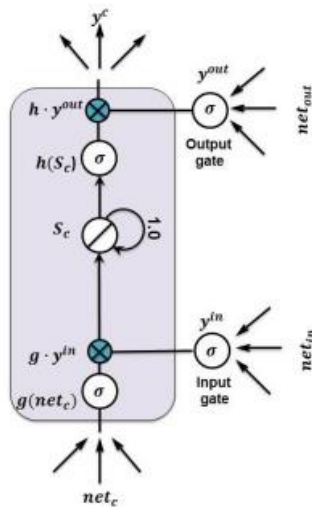
Τη λύση στο πρόβλημα της αστάθειας του εκπαιδευόμενου μοντέλου των δικτύων RNN, δίνουν τα δίκτυα Long Short Term Memory (LSTM), τα οποία αποτελούν μία υποκατηγορία των δικτύων RNN. Τα δίκτυα LSTM (Εικόνα 3.3) περιέχουν κρυφά επίπεδα με μνήμη μπλοκ με κελιά, αντί για μη γραμμικές μονάδες, και έτσι μπορούν να αποθηκεύουν πληροφορίες για μεγάλα χρονικά διαστήματα (Sak, Senior, & Beaufays, 2014).



Εικόνα 3.3: Αρχιτεκτονική LSTM RNN
Πηγή: (Sak, Senior, & Beaufays, 2014)

Τα μπλοκ μνήμης περιέχουν κελιά με αυτοσυνδέσεις, τα οποία αποθηκεύουν τη χρονική κατάσταση του δικτύου, αλλά και περιέχουν πολλαπλασιαστικές μονάδες, που ονομάζονται πύλες, και ελέγχουν τη ροή των πληροφοριών.

Μέσα σε κάθε επίπεδο του LSTM, πραγματοποιείται ένα σύνολο αριθμητικών πράξεων όπως απεικονίζεται στην Εικόνα 4.19.



Εικόνα 3.4: Παρουσίαση αριθμητικών πράξεων σε ένα κρυφό επίπεδο LSTM με δύο πύλες εισόδου
Πηγή: (Smagulova & James, 2019)

Οι «πύλες» εισόδου και εξόδου, σε κάθε κρυφό επίπεδο, αποτελούνται από πίνακες βάρους και συναρτήσεις ενεργοποίησης. Στο κέντρο ενός μπλοκ μνήμης είναι μια αυτοσυνδεόμενη γραμμική μονάδα S_c , που ονομάζεται επίσης «καρουσέλ σταθερού σφάλματος» (CEC), και προστατεύει το LSTM από την εξαφάνιση και την έκρηξη προβλημάτων διαβάθμισης (Smagulova & James, 2019).

3.3.5 Convolutional Neural Network (CNN)

Τα Convolutional Neural Networks (CNNs) χρησιμοποιούνται για την κατανόηση εικόνων, ή γενικότερα χωρικά εξαρτώμενων πληροφοριών, και μπορούν να ταξινομήσουν εικόνες, αλλά και να εξαγάγουν χαρακτηριστικά από αυτές. Τα δίκτυα CNN χρησιμοποιούν ένα εξειδικευμένο είδος γραμμικής λειτουργίας, τη συνέλιξη (convolution). Μια τυπική αρχιτεκτονική των CNN αποτελείται από το στρώμα συνέλιξης, το στρώμα συγκέντρωσης και το πλήρως συνδεδεμένο στρώμα. Πιο αναλυτικά, η αρχιτεκτονική ενός δικτύου CNN αποτελείται από τα εξής επίπεδα:

1. Συνελικτικό επίπεδο (convolutional layer)

Το συνελικτικό επίπεδο είναι πάντα το πρώτο επίπεδο αρχιτεκτονικής ενός δικτύου CNN. Η είσοδος αυτού του επιπέδου είναι τρισδιάστατο αντικείμενο. Ο σκοπός του

συνελκτικού επιπέδου, είναι να μετασχηματίσει μια εικόνα $M \times N$ διάστατη εικόνα 1 (ασπρόμαυρη) ή 3 (έγχρωμη) χαρακτηριστικών ανά pixel, σε μια 1×1 διάστατη εικόνα K χαρακτηριστικών. Μέσω αυτού του επιπέδου, το δίκτυο παίρνει το χάρτη ενεργοποίησης, ή αλλιώς χάρτη χαρακτηριστικών (Bhandare & Kaur, 2021).

2. Μη γραμμικό επίπεδο ενεργοποίησης (activation layer)

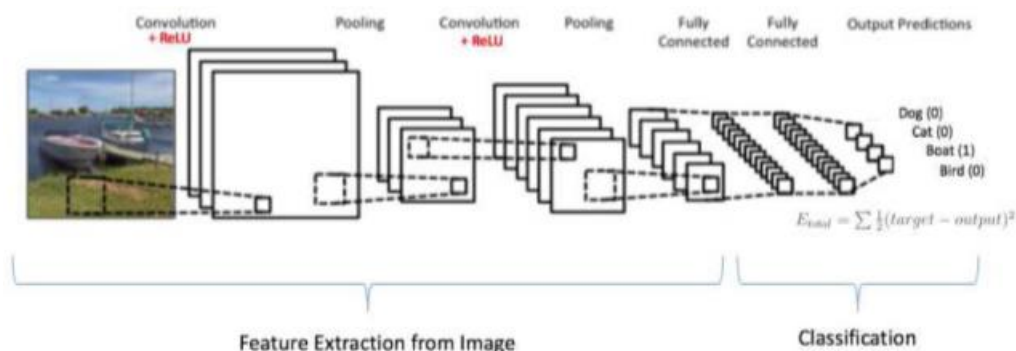
Έπειτα από τη διαδικασία συνέλιξης, οι χάρτες χαρακτηριστικών εξόδου διέρχονται στο μη γραμμικό επίπεδο ενεργοποίησης ως μια μη γραμμική συνάρτηση (Bhandare & Kaur, 2021).

3. Pooling layer

Πρόκειται για μια διαδικασία λήψης και συρρίκνωσης εικόνων, χωρίς αυτές να χάνουν τις σημαντικές τους ιδιότητες. Τα πλεονεκτήματα αυτής της διαδικασίας είναι ότι η ποσότητα των παραμέτρων και των βαρών μειώνεται κατά 75%, με συνέπεια τη μείωση του κόστους υπολογισμού, καθώς επίσης, λόγω της μείωσης του αριθμού των παραμέτρων και των υπολογισμών στο δίκτυο, ελέγχεται η υπερπροσαρμογή (Bhandare & Kaur, 2021).

4. Πλήρως συνδεδεμένο επίπεδο (fully connected layer)

Το πλήρως συνδεδεμένο επίπεδο, είναι ένα επίπεδο Perceptron, που χρησιμοποιεί μια συνάρτηση ενεργοποίησης softmax στο επίπεδο εξόδου. Η έξοδος της συνάρτησης αυτής ισοδυναμεί με μία κατηγορική συνάρτηση πιθανότητας, δίνοντας την πιθανότητα οποιαδήποτε από τις κλάσεις να είναι αληθής (Bhandare & Kaur, 2021).



Εικόνα 3.5: Παράδειγμα αρχιτεκτονικής CNN

Πηγή: (Bhandare & Kaur, 2021)

Στην Εικόνα 3.4 απεικονίζεται ένα παράδειγμα της αρχιτεκτονικής CNN, με όλα τα επίπεδα της, έχοντας ως συνάρτηση ενεργοποίησης τη συνάρτηση Relu, έπειτα από κάθε συνελικτικό επίπεδο.

3.4 Τεχνικές εκπαίδευσης

3.4.1 Γενικά

Η εκμάθηση και η εκπαίδευση των νευρωνικών δικτύων είναι δύο επαναληπτικές διαδικασίες διαμερισμού σφάλματος. Στη διαδικασία της εκμάθησης, το δείγμα πάει από τα πρώτα στα τελευταία επίπεδα, και έτσι συσσωρεύεται το σφάλμα.

Στη συνέχεια, πραγματοποιείται η αντίθετη πορεία, αλλάζοντας τα βάρη βάσει της παραγώγου. Στόχος της εκπαίδευσης είναι η ελαχιστοποίηση του μαθησιακού σφάλματος.

Η επιλογή της κατάλληλης τεχνικής εκπαίδευσης, γίνεται ανάλογα με τη μορφή του ζητούμενου μοντέλου πρόβλεψης, ή αλλιώς με τη μορφή του επιπέδου εξόδου. Οι δύο βασικότερες κατηγορίες εκπαίδευσης είναι η παλινδρόμηση (regression) και η ταξινόμηση (classification).

3.4.2 Τεχνική παλινδρόμησης (regression)

3.4.2.1 Γενικά

Η τεχνική της παλινδρόμησης (regression), χρησιμοποιείται για την πρόβλεψη ενός νέου συντελεστή, ως συνεχής τιμή. Ένα παράδειγμα είναι η πρόβλεψη της μελλοντικής αξίας των ακινήτων σε μία περιοχή, με βάση την εξέλιξη της αγοράς, ή η πρόβλεψη του μελλοντικού δείκτη IRI μιας οδού, στην οποία βασίζεται και το αντικείμενο αυτής της διπλωματικής.

Στην αναγνώριση συστήματος, η εξαρτημένη μεταβλητή y , είναι η έξοδος του συστήματος, ενώ οι ανεξάρτητες μεταβλητές x είναι οι εισόδους του συστήματος. Για την αναγνώριση του συστήματος, είναι απαραίτητη η θεώρηση μιας συνάρτησης ενεργοποίησης α , η οποία υπολογίζεται ως συνάρτηση των μεταβλητών εισόδου πολλαπλασιασμένων με τα κατάλληλα βάρη, συν μιας μεταβλητής μεροληψίας b . Επομένως, η μορφή μιας συνάρτησης ενεργοποίησης α_1 , θα έχει τη μορφή:

$$a_1 = f_1(w_1 \cdot x_1 + w_2 \cdot x_2 + w_i \cdot x_i + \dots + b_1) \quad (3.1)$$

Όπου a_1 : συνάρτηση ενεργοποίησης

x_i : μεταβλητή εισόδου

w_i : βάρος της μεταβλητής x_i

Η έξοδος y , είναι μια γραμμική συνάρτηση των συναρτήσεων ενεργοποίησης a πολλαπλασιασμένων με τα κατάλληλα βάρη w' , συν μια μεταβλητή μεροληψίας b' . Δηλαδή έχει τη μορφή:

$$y_1 = f'_1(w'_1 \cdot a_1 + w'_i \cdot a_i + \dots + b'_1) \quad (3.2)$$

Όπου y_1 : η έξοδος της συνάρτησης εκπαίδευσης

a_i : συνάρτηση ενεργοποίησης

w'_i : βάρος της συνάρτησης ενεργοποίησης a_i

b'_1 : μεταβλητή μεροληψίας

Ο υπολογισμός του σφάλματος προκύπτει με τη βοήθεια της μεθόδου των ελαχίστων τετραγώνων (Mean Squared Error -MSE), ή της μεθόδου του μέσου απόλυτου σφάλματος (Mean Absolute Error -MAE), ή άλλων μεθόδων, με πιο γνωστή την πρώτη. Η μέθοδος των ελαχίστων τετραγώνων, παρουσιάζεται από τον τύπο:

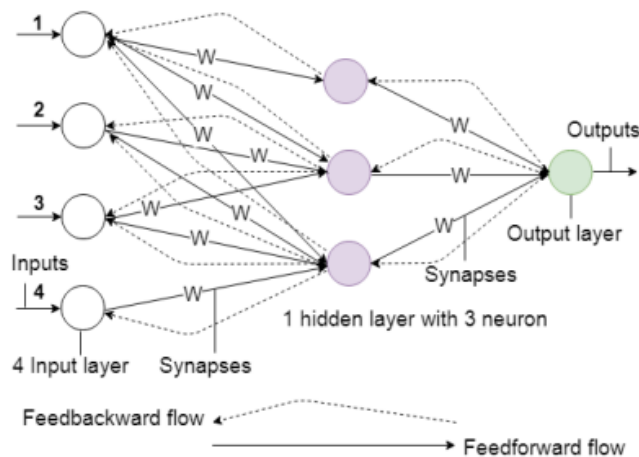
$$MSE = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} (y - y')^2 \quad (3.3)$$

Όπου $N_{samples}$: ο αριθμός των δειγμάτων

y : ο δείκτης εξόδου που προκύπτει από το μοντέλο πρόβλεψης

y' : ο πραγματικός δείκτης εξόδου

Στη συνέχεια, το σφάλμα διαμοιράζεται στα βάρη και η διαδικασία επαναλαμβάνεται, μέχρι να πραγματοποιηθεί η ελαχιστοποίηση του σφάλματος (Εικόνα 3.5).

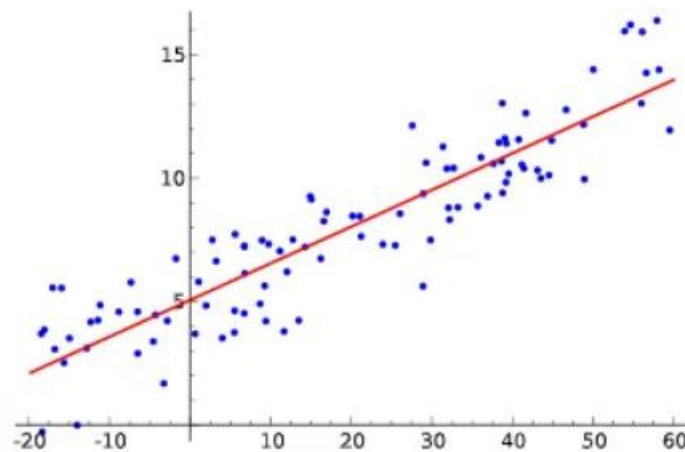


Εικόνα 3.6: Αρχιτεκτονική μοντέλου με τη μέθοδο της παλινδρόμησης
Πηγή: (Abiodun, et al., 2019)

Τα διάφορα μοντέλα της παλινδρόμησης, κατανέμονται σε κατηγορίες όπως είναι τα γραμμικά μοντέλα, τα πολλαπλά γραμμικά μοντέλα ή τα μη γραμμικά. Τα πιο σύνηθες είναι τα γραμμικά και τα πολλαπλά γραμμικά μοντέλα, καθώς τα μη γραμμικά μοντέλα χρησιμοποιούνται σε πιο πολύπλοκα προβλήματα. Τρεις βασικές κατηγορίες γραμμικών μοντέλων οι οποίες αξίζει να αναφερθούν, που είναι ίσως και οι πιο συνηθισμένες, είναι οι Linear Regression, Ridge Regression και Lasso Regression.

3.4.2.2 Linear Regression

Η Linear Regression, είναι μία γραμμική συνάρτηση που ακολουθεί της διαδικασίας της παλινδρόμησης. Μέσω αυτής της γραμμικής συνάρτησης, προκύπτει μία σχέση μεταξύ ανεξάρτητων και εξαρτημένων μεταβλητών, με τη μέθοδο των ελαχίστων τετραγώνων (σχέση 3.4).



Σχήμα 3.1: Παράδειγμα Linear Regression με μία ανεξάρτητη μεταβλητή
 Πηγή: (HP, 2019)

Στο Σχήμα 3.1 απεικονίζεται ένα απλό παράδειγμα της γραμμικής συνάρτησης Linear Regression. Ο οριζόντιος άξονας x' αντιστοιχεί στις ανεξάρτητες μεταβλητές, ενώ ο κατακόρυφος άξονας y' αντιστοιχεί στις εξαρτημένες μεταβλητές.

Τα σημεία στο διάγραμμα αντιστοιχούν στις πραγματικές τιμές του πειράματος. Η κάθετη απόσταση κάθε σημείου από τη γραμμή, αντιστοιχεί στο σφάλμα μεταξύ της αντίστοιχης προβλεπόμενης και πραγματικής τιμής. Η γραμμική σχέση, η οποία απεικονίζεται με κόκκινο χρώμα, αντιστοιχεί στις προβλεπόμενες τιμές, και είναι η ελάχιστη δυνατή απόσταση από όλες τις πραγματικές τιμές.

$$\sum_{i=1}^n (y - \hat{y})^2 \quad (3.4)$$

$$\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_n \cdot x_n \quad (3.5)$$

Όπου y : η πραγματική τιμή

$\hat{y}$: η τιμή της πρόβλεψης

x_n : η ανεξάρτητη μεταβλητή

b_n : η κλίση της ευθείας των προβλεπόμενων $\hat{y}$

Σημειώνεται ότι, στο συγκεκριμένο παράδειγμα της εικόνας, υπάρχει μόνο μία ανεξάρτητη μεταβλητή, για αυτό και η σχέση είναι γραμμική. Η σχέσεις μεταξύ εξαρτημένων και ανεξάρτητων μεταβλητών μπορεί να είναι και μη γραμμική, όπως στην παρούσα διπλωματική εργασία, όπου οι ανεξάρτητες μεταβλητές είναι περισσότερες από μία. Σε αυτή την περίπτωση, θα απεικονιζόταν με μία καμπύλη γραμμή.

Τέλος, η σχέση μεταξύ εξαρτημένης και ανεξάρτητης μεταβλητής, μπορεί να είναι είτε θετική, είτε αρνητική. Η διαφορά μεταξύ θετικής και αρνητικής σχέσης των δύο μεταβλητών, είναι ότι στη θετική σχέση η κλίση της ευθείας είναι θετική, ενώ στην αρνητική σχέση η κλίση είναι αρνητική.

3.4.2.3 Ridge Regression

Η Ridge Regression είναι μία προέκταση της Linear Regression, όπου η συνάρτηση τροποποιείται με στόχο την αποφυγή της υπερπροσαρμογής (overfitting). Η υπερπροσαρμογή δημιουργείται όταν το μοντέλο αποδίδει καλά στα δεδομένα εκπαίδευσης, αλλά δεν έχει καλή απόδοση στις δοκιμές (testing).

Ένα ιδανικό μοντέλο θα ήταν αυτό που αποδίδει εξίσου καλά στα δεδομένα εκπαίδευσης, τόσο όσο και στις δοκιμές. Για το σκοπό αυτό, στη συνάρτηση ελαχιστοποίησης των τετραγώνων, της Linear Regression, προστίθεται μία ακόμα παράμετρος (σχέση 3.6).

$$\sum_{i=1}^n (y - \hat{y})^2 + \lambda \cdot slope^2 \quad (3.6)$$

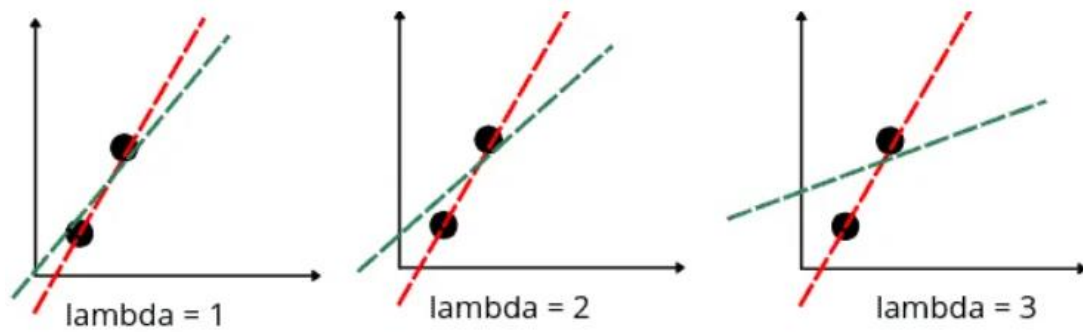
Όπου y : η πραγματική τιμή

$\hat{y}$: η τιμή της πρόβλεψης

λ : θετική παράμετρος $\in (0, +\infty)$

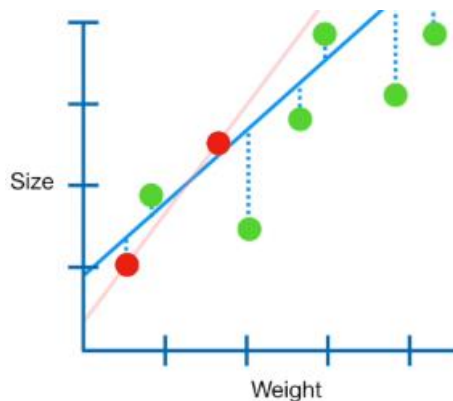
slope: η κλίση της ευθείας των προβλεπόμενων $\hat{y}$

Όσο μεγαλώνει η παράμετρος λ , τόσο μειώνεται η κλίση της ευθείας των προβλεπόμενων τιμών (Σχήμα 3.2), αλλά ποτέ δε θα μηδενιστεί.



Σχήμα 3.2: Ridge Regression για διάφορες τιμές του λ
Πηγή: (Alam, 2022)

Επομένως, μέσω της Ridge Regression, μειώνεται η κλίση της ευθείας των προβλεπόμενων $\hat{y}$, επιτυγχάνοντας τη βελτίωση της διακύμανσης μεταξύ των δεδομένων εκπαίδευσης και των δοκιμών, αποφεύγοντας την υπερπροσαρμογή (Σχήμα 3.3).



Σχήμα 3.3: Μοντέλο με υπερπροσαρμογή (overfitting)
Πηγή: (Perlato, 2019)

Στο Σχήμα 3.3, απεικονίζονται με κόκκινο χρώμα τα δεδομένα εκπαίδευσης, ενώ με πράσινο τα δεδομένα δοκιμών (testing). Όπως φαίνεται, λαμβάνοντας υπόψη τη γραμμική σχέση μεταξύ των δεδομένων εκπαίδευσης (κόκκινη γραμμή), το μοντέλο αποδίδει καλά μόνο στα δεδομένα εκπαίδευσης και όχι στις δοκιμές (υπερπροσαρμογή). Αντιθέτως, με τη βοήθεια της συνάρτησης των ελάχιστων τετραγώνων της Ridge Regression, δημιουργείται μία νέα σχέση (μπλε γραμμή), η οποία έχει ικανότητα γενίκευσης.

3.4.2.4 Lasso Regression

Η συνάρτηση παλινδρόμησης Lasso Regression, μοιάζει πολύ με το μοντέλο της Ridge Regression, με μια βασική διαφορά ως προς τον τύπο των ελαχίστων τετραγώνων (σχέση 3.7).

$$\sum_{i=1}^n (y - \hat{y})^2 + \lambda \cdot |slope| \quad (3.7)$$

Όπου y : η πραγματική τιμή

$\hat{y}$: η τιμή της πρόβλεψης

λ : θετική παράμετρος $\in (0, +\infty)$

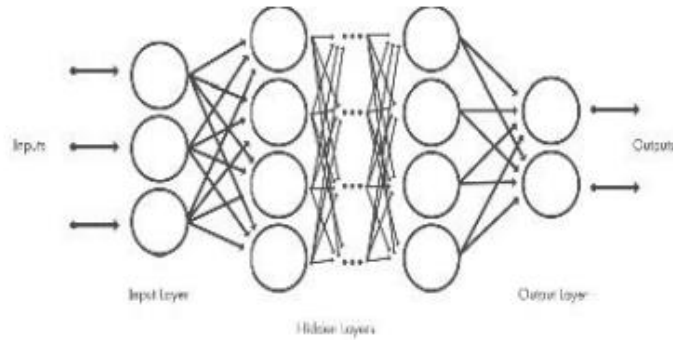
slope: η κλίση της ευθείας των προβλεπόμενων $\hat{y}$

Σε αυτή την περίπτωση η κλίση (slope) εμφανίζεται ως μέγεθος. Αυτό έχει ως αποτέλεσμα μετά από κάποιες επαναλήψεις, κάποια στοιχεία, τα οποία έχουν μικρή συσχέτιση με την τιμή πρόβλεψης του ζητούμενου, να μηδενίζονται. Έτσι, όσο αυξάνεται το λ η κλίση μειώνεται, προσεγγίζοντας τη μηδενική κλίση, που κάποια στιγμή ίσως να γίνει και μηδενική.

Συνεπώς, σε προβλήματα όπου οι μεταβλητές σχετίζονται άμεσα με το ζητούμενο y , η Ridge Regression δίνει καλύτερα αποτελέσματα. Αντίθετα, σε προβλήματα όπου κάποιες μεταβλητές πιθανώς να μη συσχετίζονται άμεσα με το ζητούμενο y , η Lasso Regression θα παρουσιάζει μικρότερα σφάλματα, καθώς το βάρος αυτών των μεταβλητών θα μηδενιστεί.

3.4.3 Τεχνική ταξινόμησης (classification)

Η τεχνική της ταξινόμησης (classification), χρησιμοποιείται για την πρόβλεψη μιας διακριτής κατηγορίας. Ένα απλό παράδειγμα, το οποίο επιλύεται με τη μέθοδο της ταξινόμησης, είναι η πρόβλεψη και η κατηγοριοποίηση της ηλικίας ενός ανθρώπου ανά δεκαετία, σύμφωνα με διάφορα χαρακτηριστικά.



Εικόνα 3.7: Αρχιτεκτονική μοντέλου με τη μέθοδο της ταξινόμησης
 Πηγή: (Ruby, Theerthagiri, Jacob, & Dr.Y.Vamsidhar, 2020)

Στην Εικόνα 3.6 απεικονίζεται ένα παράδειγμα αρχιτεκτονικής μοντέλου εκπαιδευμένο με τη μέθοδο της ταξινόμησης, το οποίο έχει ως δεδομένο εξόδου την κατάταξη των ζητούμενων σε δύο ή περισσότερες διακριτές κατηγορίες.

Στο μοντέλο της ταξινόμησης, κάθε στοιχείο παρουσιάζεται ως συνδυασμός των αριθμών 0 και 1. Σε αυτή την περίπτωση, η συνάρτηση ενεργοποίησης είναι η μη γραμμική συνάρτηση «softmax» (σχέση 3.8), η οποία είναι γνωστή ως κανονικοποιημένη εκθετική συνάρτηση, και μετατρέπει ένα διάνυσμα πραγματικών αριθμών σε κατανομή πιθανότητας. Κάθε δείκτης εξόδου y , μπορεί να πάρει τιμές μεταξύ 0 και 1, αφού πρόκειται για πιθανότητα. Τέλος, το αποτέλεσμα της πρόβλεψης του δικτύου θα είναι η μέγιστη τιμή των δεικτών εξόδου y (σχέση 3.9).

$$y_i = \text{softmax}(\sum(w_i a_i + b_i)) , 0 \leq y \leq 1 \quad (3.8)$$

$$\text{Network prediction} \rightarrow \max(y_i) \quad (3.9)$$

Όπου y_i : η έξοδος της συνάρτησης εκπαίδευσης

softmax : μη γραμμική συνάρτηση ενεργοποίησης

a_i : συνάρτηση ενεργοποίησης

w_i : βάρος της συνάρτησης ενεργοποίησης a_i

b_i : μεταβλητή μεροληψίας

Η πιο γνωστή μέθοδος για τον υπολογισμό σφάλματος στη μέθοδο της ταξινόμησης, είναι η «Cross-Entropy». Το σφάλμα που υπολογίζεται από τον τύπο (σχέση 3.10)

του CE, μεταξύ της κατηγορίας πρόβλεψης του δικτύου y_k και του πραγματικού αποτελέσματος y'_k , διαμοιράζεται στα βάρη και η διαδικασία επαναλαμβάνεται, μέχρι το σφάλμα να ελαχιστοποιηθεί.

$$CE = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} \sum_{k=1}^{N_{categories}} y_k \log(y'_k) \quad (3.10)$$

Όπου $N_{samples}$: ο αριθμός των δειγμάτων

$N_{categories}$: ο αριθμός των κατηγοριών

y_k : η κατηγορία που προκύπτει από το μοντέλο πρόβλεψης

y'_k : η πραγματική κατηγορία

4. ΑΝΑΛΥΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

4.1 Συνοπτική περιγραφή

Όπως αναφέρθηκε στην Εισαγωγή της παρούσας διπλωματικής εργασίας, στόχος είναι η παρουσίαση των Νευρωνικών Δικτύων, ως ένα μέσο πρόβλεψης της ομαλότητας των οδοστρωμάτων. Για να «ελεγχθεί» η αξιοπιστία των Νευρωνικών Δικτύων, ως προς την πρόβλεψη των μελλοντικών δεικτών IRI, πραγματοποιούνται τέσσερις πειραματικές αναλύσεις, με χρήση της γλώσσας προγραμματισμού Python και της βιβλιοθήκης pytorch.

Αξιοποιούνται στοιχεία από δύο τμήματα αυτοκινητοδρόμου, προερχόμενα από τη βάση δεδομένων του Εργαστηρίου Οδοστρωμάτων του ΕΜΠ. Τα δύο αυτά τμήματα, έστω τμήματα Α και Β, αποτελούνται από τρεις λωρίδες το κάθε ένα και έχουν μήκους τρία χιλιόμετρα αντίστοιχα. Αναλυτικότερα, αξιοποιούνται οι δείκτες IRI για τα δύο τμήματα, για τα έτη 2013 έως και 2021. Οι μετρήσεις, για κάθε τμήμα, έχουν πραγματοποιηθεί ανά 10 μέτρα για τη δεξιά (Λ2) και την αριστερή (Λ1) λωρίδα της οδού. Σε κάθε λωρίδα έχουν γίνει μετρήσεις για τα δύο ίχνη, αριστερά (L) και δεξιά (R). Έτσι, για παράδειγμα, η μέτρηση για το δεξί ίχνος της αριστερής λωρίδας, συμβολίζεται ως R(Λ1). Επιπλέον, αξιοποιούνται τα στοιχεία παχών του ασφαλτικού οδοστρώματος για κάθε χιλιομετρική θέση στα δύο τμήματα του αυτοκινητοδρόμου. Τέλος, αξιοποιήθηκαν τα διατιθέμενα στοιχεία κυκλοφορίας, για τα οποία γίνεται κατάλληλη επεξεργασία.

Οι τέσσερις πειραματικές αναλύσεις, έστω 1, 2, 3, και 4, πραγματοποιούνται με χρήση πέντε διαφορετικών μοντέλων μελλοντικών προβλέψεων, εκ των οποίων τα τρία είναι γραμμικές συναρτήσεις, ενώ τα άλλα δύο είναι Νευρωνικά Δίκτυα.

Σε κάθε πειραματική ανάλυση, ως δεδομένα εισόδου ορίζονται τα δεδομένα του τμήματος Α, με τα οποία και εκπαιδεύονται τα δίκτυα. Στις πειραματικές αναλύσεις 1 και 2, πραγματοποιείται η πρόβλεψη των δεικτών IRI, για τις δύο λωρίδες (αριστερής και δεξιάς αντίστοιχα) του τμήματος Α, για το έτος 2021. Ομοίως, για τις πειραματικές αναλύσεις 3 και 4, πραγματοποιείται η πρόβλεψη των δεικτών IRI του τμήματος Β του αυτοκινητοδρόμου, για το ίδιο έτος. Έτσι, με τη σύγκριση των μελλοντικών δεικτών IRI, που προκύπτουν από τις πειραματικές αναλύσεις, με τους πραγματικούς δείκτες IRI, από τη βάση δεδομένων, προκύπτουν συμπεράσματα ως προς την ικανότητα των Νευρωνικών Δικτύων να προβλέπουν με επιτυχία την

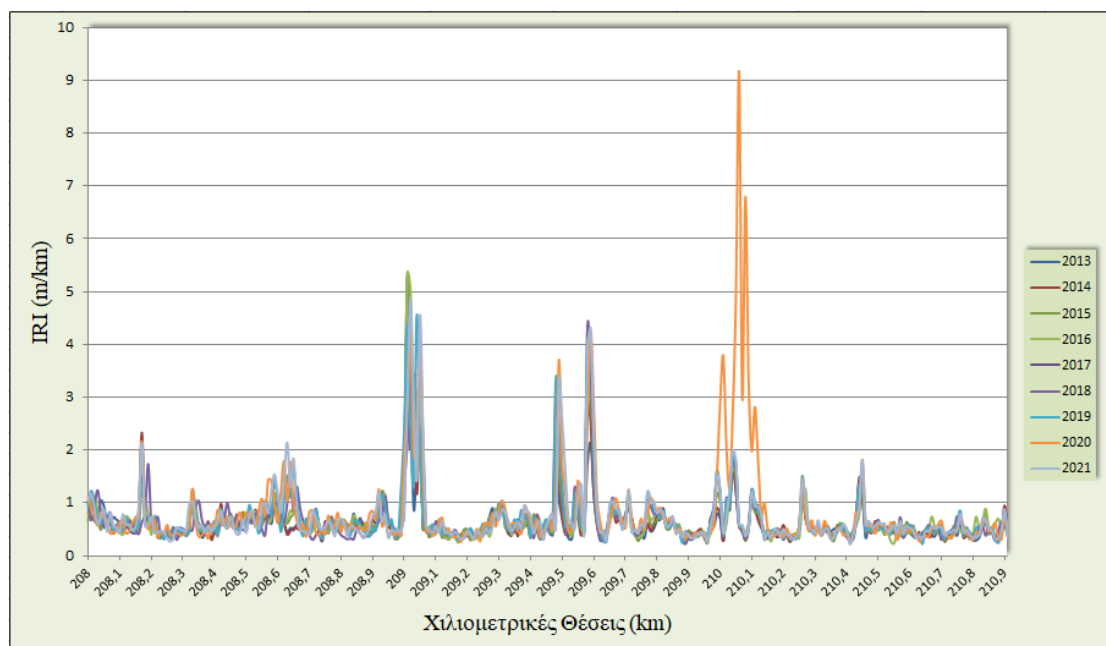
μελλοντική ομαλότητα του οδοστρώματος μίας οδού, αλλά και συμπεράσματα γενίκευσης των Νευρωνικών Δικτύων.

Τέλος, με τον προσδιορισμό του σφάλματος των μεθόδων πρόβλεψης, σε κάθε μία από τις προαναφερθείσες πειραματικές αναλύσεις, προσδιορίζεται η καλύτερη μέθοδος πρόβλεψης της ομαλότητας, συγκρίνοντας τα αποτελέσματα που προκύπτουν από τις γραμμικές συναρτήσεις με αυτά που προκύπτουν από τα Νευρωνικά Δίκτυα.

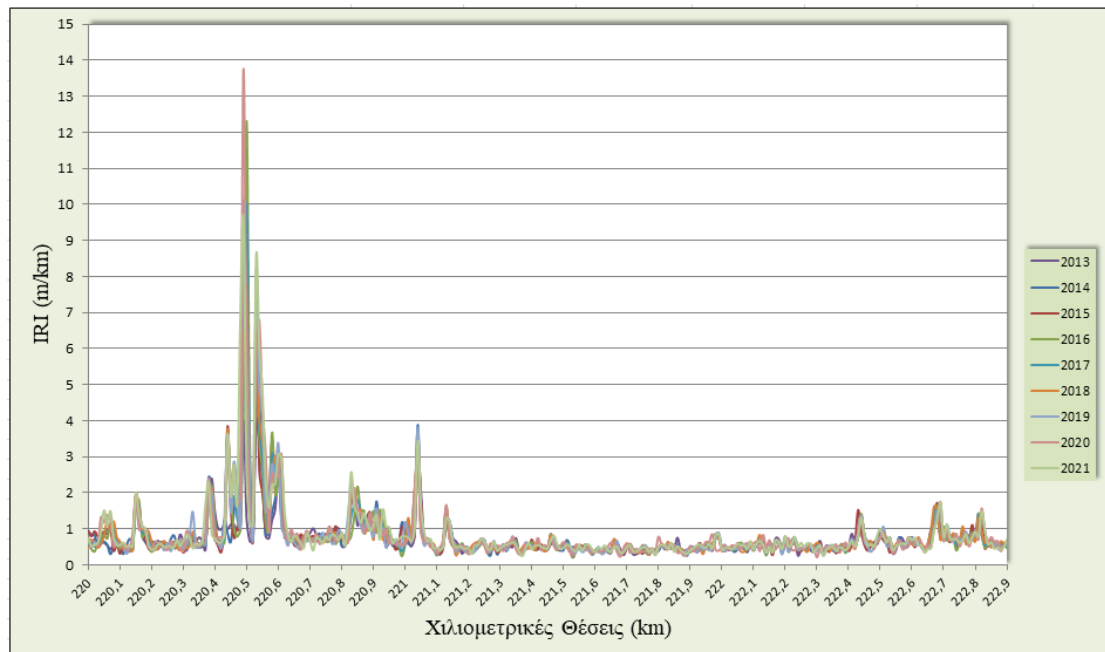
4.2 Συλλογή στοιχείων

4.2.1 Δείκτες ομαλότητας IRI

Στους παρακάτω πίνακες παρουσιάζονται οι κατανομές του δείκτη IRI, για το δεξί ίχνος της αριστερής λωρίδας R(Λ1), για το τμήμα A (Σχήμα 4.1) και το τμήμα B (Σχήμα 4.2).



Σχήμα 4.1: Δείκτες IRI για το δεξί ίχνος της αριστερής λωρίδας του τμήματος A



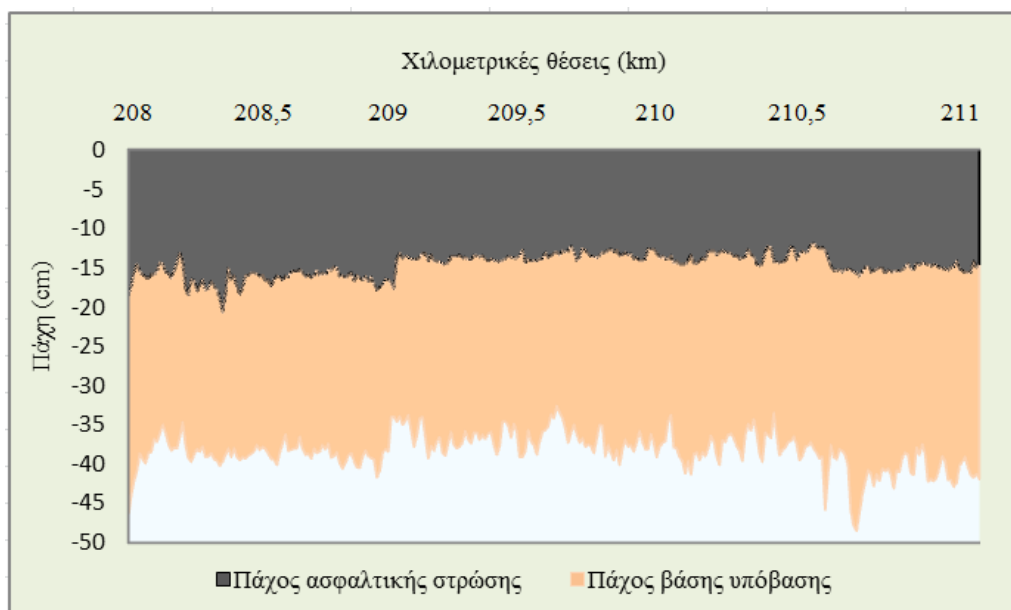
Σχήμα 4.2: Δείκτες IRI για το δεξί ίχνος της αριστερής λωρίδας του τμήματος Β

Όπως απεικονίζεται και στο Σχήμα 4.1, οι δείκτες IRI του τμήματος Α, κυμαίνονται από 0,3 έως και 9,15 , με το μεγαλύτερο εύρος να κυμαίνεται από 0,3 έως και 0,7 κατά μήκος του τμήματος. Ομοίως, οι δείκτες του τμήματος Β (Σχήμα 4.2), κυμαίνονται από 0,2 έως και 13,8 , με το μεγαλύτερο εύρος να κυμαίνεται από 0,2 έως και 0,7 κατά μήκος του τμήματος.

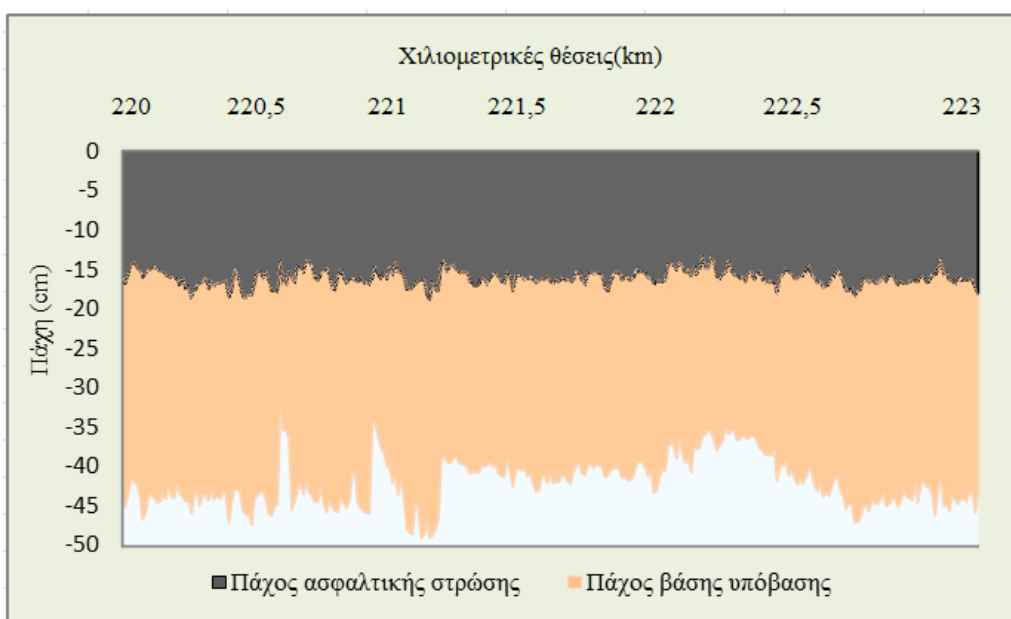
Οι κατά περίπτωση υψηλές τιμές του δείκτη IRI είναι περιορισμένες τοπικά και κρίνεται ότι δεν αντικατοπτρίζουν τη γενική λειτουργική κατάσταση του υπό διερεύνησης οδοστρώματος.

4.2.2 Πάχη οδοστρώματος

Για κάθε χιλιομετρική θέση, στα δύο τμήματα της οδού, διατίθενται τα πάχη των ασφαλτικών στρώσεων (h1), καθώς και τα πάχη της στρώσης βάση -υπόβαση (h2). Τα πάχη είναι μετρημένα με το σύστημα Ground Penetrating Radar (GPR) και αποτελούν στοιχεία από τη βάση δεδομένων του Εργαστηρίου Οδοστρωμάτων του ΕΜΠ.



Σχήμα 4.3: Πάχη τμήματος Α ανά χιλιομετρική θέση



Σχήμα 4.4: Πάχη τμήματος Β ανά χιλιομετρική θέση

Όπως απεικονίζεται και στα Σχήματα 4.3 και 4.4, τα πάχη της ασφαλτικής στρώσης των δύο τμημάτων της οδού κυμαίνονται από 12 έως 20 cm, ενώ τα πάχη της βάσης – υπόβασης κυμαίνονται από 16 έως 33 cm, με τις τιμές αυτές να μη διαφέρουν κατά πολύ στα δύο αυτά τμήματα.

4.2.3 Στοιχεία κυκλοφορίας

Τέλος, συλλέχθηκαν τα διατιθέμενα στοιχεία κυκλοφορίας, για τα δύο τμήματα της οδού, για τα έτη 2012 έως και 2021 (Πίνακας 4.1).

Πίνακας 4.1: ΕΜΗΚ των τμημάτων Α και Β

Χ.Θ./ ΕΤΗ	2012			2013		
	AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	2.337	2.143	194	2.994	2.728	266
220-222,9	1.720	1.624	96	2.466	2.296	170
	2014			2015		
	AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	2.825	2.574	251	3.073	2.806	267
220-222,9	2.432	2.264	168	2.461	2.289	172
	2016			2017		
	AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	3.015	2.752	262	3.176	2.900	276
220-222,9	2.678	2.491	187	2.807	2.625	182
	2018			2019		
	AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	3.205	2.916	289	3.342	3.053	289
220-222,9	2.859	2.665	194	2.991	2.789	202
	2020			2021		
	AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	2.593	2.344	249	2.865	2.595	270
220-222,9	2.332	2.108	224	2.627	2.380	247

Στα στοιχεία αυτά περιλαμβάνεται η ΕΜΗΚ για τα έτη 2012-2021, καθώς και ο διαχωρισμός των οχημάτων σε ελαφρά οχήματα (Light Vehicles – LV) και βαρέα οχήματα (Heavy Vehicles – HV).

Παρατηρείται ότι το τμήμα Α της οδού (208-210,9), έχει μεγαλύτερη ετήσια κυκλοφορία από ότι το τμήμα Β (220-222,9), τόσο στα βαρέα όσο και στα ελαφριά οχήματα, κάτι το οποίο αιτιολογεί μερικώς τη διαφορά στο εύρος δεικτών IRI μεταξύ των δύο τμημάτων. Παρόλα αυτά, η διαφορά της ετήσιας κυκλοφορίας μεταξύ των δύο οδών, δεν είναι σημαντική.

Πριν την έναρξη της αναλυτικής διαδικασίας, έγινε η μετατροπή των στοιχείων κυκλοφορίας από ΕΜΗΚ σε ΙΤΑ, με τον πολλαπλασιασμό των βαρέων οχημάτων (Heavy Vehicles – HV) με τον συντελεστή 1,2 , σε κάθε έτος. Έπειτα, οι μετρήσεις των

στοιχείων κυκλοφορίας αναφέρονται στη διέλευση των οχημάτων σε το μήκος της οδού και όχι τη συνολική διέλευση στις λωρίδες στις οποίες έχουν πραγματοποιηθεί οι μετρήσεις. Έτσι, έγιναν οι εξής παραδοχές, με στόχο τη διαμοίραση των οχημάτων διέλευσης στη δεξιά και αριστερή λωρίδα της οδού.

Με τη θεώρηση ότι τα τμήματα της οδού αποτελούνται από τρεις λωρίδες το κάθε ένα, έγιναν οι παρακάτω υποθέσεις. Έστω ότι στην αριστερή λωρίδα (L), λόγω της χρήσης της για προσπεράσεις και απαγόρευσης της διέλευσης βαρέων οχημάτων σε αυτή, διέρχεται το 1% των συνολικών βαρέων οχημάτων και το 25% των συνολικών ελαφριών οχημάτων. Και έστω ότι στη δεξιά λωρίδα (R), λόγω της χρήσης της από την συντριπτική πλειοψηφία των βαρέων οχημάτων, διέρχεται το 94% των συνολικών βαρέων οχημάτων και το 35% των συνολικών ελαφριών οχημάτων. Έτσι προέκυψαν τα στοιχεία κυκλοφορίας, σε ΙΤΑ, για τις δύο λωρίδες διερεύνησης, για κάθε έτος, όπως αναγράφονται στον Πίνακα 4.2.

Πίνακας 4.2: ΙΤΑ ανά λωρίδα για τα τμήματα Α και Β

Χ.Θ./ΕΤΗ	ΛΩΡΙΔΑ	2012			2013		
		AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	L	538	536	2	685	682	3
	R	969	750	219	1255	955	301
220-222,9	L	407	406	1	576	574	2
	R	677	568	109	995	804	192
		2014			2015		
		AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	L	646	643	3	705	701	3
	R	1184	901	284	1284	982	302
220-222,9	L	568	566	2	574	572	2
	R	982	792	189	995	801	194
		2016			2017		
		AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	L	691	688	3	728	725	3
	R	1259	963	296	1327	1015	312
220-222,9	L	625	623	2	658	656	2
	R	1083	872	211	1124	919	206
		2018			2019		
		AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV
208-210,9	L	732	729	3	767	763	3
	R	1347	1020	326	1394	1069	326
220-222,9	L	668	666	2	700	697	2
	R	1152	933	219	1204	976	227
		2020			2021		
		AADT TOTAL	AADT LV	AADT HV	AADT TOTAL	AADT LV	AADT HV

208-210,9	L	589	586	3	652	649	3
	R	1101	820	281	1212	908	304
220-222,9	L	530	527	3	598	595	3
	R	990	738	253	1112	833	279

Σημειώνεται ότι η διαφορά του αθροίσματος των προαναφερθέντων ποσοστών από το συνολικό διερχόμενο 100%, θεωρείται ότι διέρχεται από τη μεσαία λωρίδα, η οποία δε διερευνάται στην παρούσα διπλωματική εργασία.

4.3 Πλαίσιο πειραματικών αναλύσεων

Η διαδικασία του προγραμματισμού ξεκινάει με τη δημιουργία ενός αρχείου «utils.py», όπου εισάγονται όλα τα δεδομένα και διαχωρίζονται σε στοιχεία εισόδου και εξόδου.

Στη συνέχεια, ορίζονται οι τέσσερις πειραματικές αναλύσεις, όπως προαναφέρθηκε στο υποκεφάλαιο 4.1 (Εικόνα 4.1).

```

34 experiments = {
35     1: {'break_type': 'percent',
36         'dataset':
37             {'kms': ['208-211'], 'lane': ['1'], 'pos': ['L', 'R']}},
38     2: {'break_type': 'percent',
39         'dataset':
40             {'kms': ['208-211'], 'lane': ['2'], 'pos': ['L', 'R']}},
41     3: {'break_type': 'set',
42         'train': {'kms': ['208-211'], 'lane': ['1'], 'pos': ['L', 'R']},
43         'test': {'kms': ['220-223'], 'lane': ['1'], 'pos': ['L', 'R']}},
44     4: {'break_type': 'set',
45         'train': {'kms': ['208-211'], 'lane': ['2'], 'pos': ['L', 'R']},
46         'test': {'kms': ['220-223'], 'lane': ['2'], 'pos': ['L', 'R']}}
47 }
```

Εικόνα 4.1: Ορισμός πειραματικών αναλύσεων

Οι τέσσερις πειραματικές αναλύσεις, παρουσιάζονται αναλυτικά ακολούθως.

Με εκπαίδευση του δικτύου για τα δεδομένα του τμήματος Α (208-210,9), για τα έτη από 2013 έως και 2020, πραγματοποιείται η πρόβλεψη για το έτος 2021 των δεικτών IRI της αριστερής λωρίδας (Λ1). Στη συνέχεια γίνεται η σύγκριση των αποτελεσμάτων με τους πραγματικούς δείκτες IRI για το ίδιο έτος, της αντίστοιχης λωρίδας (Πειραματική ανάλυση 1).

Με εκπαίδευση του δικτύου για τα δεδομένα του τμήματος A (208-210,9), για τα έτη από 2013 έως και 2020, πραγματοποιείται η πρόβλεψη για το έτος 2021 των δεικτών IRI της δεξιάς λωρίδας (Λ2). Στη συνέχεια γίνεται η σύγκριση των αποτελεσμάτων με τους πραγματικούς δείκτες IRI για το ίδιο έτος, της αντίστοιχης λωρίδας (Πειραματική ανάλυση 2).

Με εκπαίδευση του δικτύου για τα δεδομένα του τμήματος A (208-210,9), για τα έτη από 2013 έως και 2020, πραγματοποιείται η πρόβλεψη, για το έτος 2021, των δεικτών IRI της αριστερής λωρίδας (Λ1) του τμήματος B (220-222,9). Στη συνέχεια γίνεται η σύγκριση των αποτελεσμάτων με τους πραγματικούς δείκτες IRI για το ίδιο έτος, της αντίστοιχης λωρίδας (Πειραματική ανάλυση 3).

Με εκπαίδευση του δικτύου για τα δεδομένα του τμήματος A (208-210,9), για τα έτη από 2013 έως και 2020, πραγματοποιείται η πρόβλεψη, για το έτος 2021, των δεικτών IRI της δεξιάς λωρίδας (Λ2) του τμήματος B (220-222,9). Στη συνέχεια γίνεται η σύγκριση των αποτελεσμάτων με τους πραγματικούς δείκτες IRI για το ίδιο έτος, της αντίστοιχης λωρίδας (Πειραματική ανάλυση 4).

4.4 Επιλογή Μοντέλου

Με στόχο την ανάδειξη των Νευρωνικών Δικτύων ως την καλύτερη μέθοδο πρόβλεψης μελλοντικών δεικτών IRI, χρησιμοποιούνται δύο κατηγορίες μεθόδων μελλοντικών προβλέψεων. Η μία κατηγορία αφορά την πρόβλεψη μέσω γραμμικών συναρτήσεων, ενώ η άλλη αφορά την πρόβλεψη μέσω Νευρωνικών Δικτύων.

Πιο συγκεκριμένα, στην κατηγορία των γραμμικών συναρτήσεων, πραγματοποιείται η πρόβλεψη μέσω των εξής τριών μεθόδων:

Γραμμική συνάρτηση 1: Linear Regression (Υποκεφάλαιο 3.4.2.2)

Γραμμική συνάρτηση 2: Ridge Regression (Υποκεφάλαιο 3.4.2.3)

Γραμμική συνάρτηση 3: Lasso Regression (Υποκεφάλαιο 3.4.2.4)

Στη δεύτερη κατηγορία, των Νευρωνικών Δικτύων, πραγματοποιείται η πρόβλεψη μέσω των εξής δύο μεθόδων:

Νευρωνικό Δίκτυο 1: Multilayer Perceptron (MLP) (Υποκεφάλαιο 3.3.2)

Νευρωνικό Δίκτυο 2: Long Short Term Memory Networks (LSTM) (Υποκεφάλαιο 3.3.4)

Για την καλύτερη δυνατή προσέγγιση των αποτελεσμάτων, στόχος είναι η μείωση του σφάλματος μεταξύ των πραγματικών δεικτών IRI και των προβλεπομένων. Συνεπώς, με την ολοκλήρωση της διαδικασίας, επιλέγεται για την ανάδειξη των αποτελεσμάτων, η μέθοδος με το μικρότερο σφάλμα μεταξύ των προβλεπομένων και των πραγματικών δεικτών, για το έτος 2021.

4.5 Γραμμικές συναρτήσεις

Όπως αναφέρθηκε και στο υποκεφάλαιο 3.4.2, η πρόβλεψη των γραμμικών συναρτήσεων με τη μέθοδο της παλινδρόμησης (regression), γίνεται με τον πολλαπλασιασμό των δεδομένων εισόδου «x train» με τα κατάλληλα βάρη w , δίνοντας ένα δεδομένο εξόδου «y train». Στη συνέχεια, μέσω της συνάρτησης των ελαχίστων τετραγώνων (MSE), η οποία διαφέρει για καθεμία από τις παρακάτω μεθόδους, υπολογίζεται το βέλτιστο μοντέλο πρόβλεψης. Το μοντέλο αυτό, με δεδομένα εισόδου τα στοιχεία «x test», δίνει την πρόβλεψη «y test».

Από τα δεδομένα συλλογής, ορίζεται το 80% των δεδομένων ως δεδομένα εκπαίδευσης (Εικόνα 4.2). Σημειώνεται ότι, ως δεδομένα εισόδου «x train», ορίζονται όλα τα στοιχεία που επηρεάζουν το δείκτη ομαλότητας IRI, δηλαδή τα στοιχεία παχών του ασφατικού οδοστρώματος και τα στοιχεία κυκλοφορίας, για τα έτη 2013 έως και 2020, ενώ ως δεδομένα εισόδου «y train», ορίζονται οι δείκτες ομαλότητας για τα έτη 2013 έως και 2020.

Το υπόλοιπο 20% των δεδομένων χρησιμοποιείται για την πρόβλεψη των δεικτών IRI. Δηλαδή, με ένα τυχαίο σύνολο δεδομένων «x test», το οποίο αποτελεί τα 20% των συνολικών δεδομένων, προκύπτουν οι δείκτες IRI για το έτος 2021 «y test».

Επιπλέον, για την καλύτερη προσέγγιση των αποτελεσμάτων, ορίζονται τα min και max των δεδομένων.

```

59 # Break in training/test data
60 break_idx = int(len(df) * 0.8)
61
62 if shuffle_test_train:
63     shuffled_df = df.sample(frac=1, random_state=seed)
64     df_train = shuffled_df[:break_idx]
65     df_test = shuffled_df[break_idx:]
66 else:
67     df_train = df[:break_idx]
68     df_test = df[break_idx:]
69
70 X_train = df_train[df.columns[:-1]]
71 y_train = df_train[df.columns[-1]]
72
73 X_test = df_test[df.columns[:-1]]
74 y_test = df_test[df.columns[-1]]
75 else:
76     traffic = extract_traffic_data(pth)
77     df, _min, _max = extract_data(pth, experiments, exp_id, _set='train', normalize=True)
78     train_min = _min
79     train_max = _max
80     if shuffle_test_train:
81         shuffled_df = df.sample(frac=1, random_state=seed)
82         df_train = shuffled_df[:]
83     else:
84         df_train = df[:]
85
86 X_train = df_train[df.columns[:-1]]
87 y_train = df_train[df.columns[-1]]
88
89 df, _min, _max = extract_data(pth, experiments, exp_id, _set='test', normalize=True)
90 test_min = _min
91 test_max = _max
92 if shuffle_test_train:
93     shuffled_df = df.sample(frac=1, random_state=seed)
94     df_test = shuffled_df[:]
95 else:
96     df_test = df[:]
97
98 X_test = df_train[df.columns[:-1]]
99 y_test = df_train[df.columns[-1]]

```

Εικόνα 4.2: Διαχωρισμός δεδομένων σε δεδομένα train και test

Επιπλέον, απαραίτητη για τη διαδικασία της παλινδρόμησης, είναι η εισαγωγή των στοιχείων x και y σε πίνακες (arrays) (Εικόνα 4.3). Τα στοιχεία x εισάγονται σε ένα δισδιάστατο πίνακα, ενώ τα στοιχεία y εισάγονται σε ένα μονοδιάστατο πίνακα. Έτσι δημιουργείται ένα μοντέλο πολλαπλής γραμμικής παλινδρόμησης.


```

102 # transform to arrays for regressors:
103 position = df_train.iloc[0]['km']
104
105 X_test_array = X_test.to_numpy()
106 y_test_array = y_test.to_numpy()
107
108 idx_start = 3
109 if use_thickness:
110     idx_start = 1
111 if use_traffic:
112     # Get train traffic
113     positions = X_train['km'].tolist()
114     years = [int('20' + y) for y in list(X_train.keys())[3:]]
115     train_traffic = []
116     for position in positions:
117         train_traffic.append(get_traffic_from_year(position, traffic, years))
118     train_traffic = np.array(train_traffic)
119     X_train_array = np.concatenate([X_train[X_train.columns[idx_start:]].to_numpy(), train_traffic], axis=1)
120     y_train_array = y_train.to_numpy()
121
122     # Get test traffic
123     positions = X_test['km'].tolist()
124     years = [int('20' + y) for y in list(X_test.keys())[3:]]
125     test_traffic = []
126     for position in positions:
127         test_traffic.append(get_traffic_from_year(position, traffic, years))
128     test_traffic = np.array(test_traffic)
129     X_test_array = np.concatenate([X_test[X_test.columns[idx_start:]].to_numpy(), test_traffic], axis=1)
130     y_test_array = y_test.to_numpy()

```

Εικόνα 4.3: Εισαγωγή στοιχείων σε πίνακες

Έχοντας εισάγει και προσαρμόσει κατάλληλα όλα τα δεδομένα, επικαλούνται τα παρακάτω μοντέλα για την πρόβλεψη των δεικτών.

4.5.1 Linear Regression

Με την εισαγωγή της συνάρτησης Linear Regression (Εικόνα 4.4) προβλέπονται, μέσω του πολλαπλασιασμού των τελικών δεικτών «x test» με τα κατάλληλα βάρη, οι δείκτες IRI για το έτος 2021 «y test».

```

148 reg = LinearRegression()
149 reg.fit(X_train_array, y_train_array)
150 y_pred = reg.predict(X_test_array)
151 print('Error for LinearRegression is:', rmse(y_pred, y_test))
152 y_pred = list(y_pred)
153 y_test_array = list(y_test_array)
154 for i in range(len(y_pred)):
155     y_pred[i] = y_pred[i]*(_max - _min)+_min
156     y_test_array[i] = y_test_array[i]*(_max - _min)+_min
157 pd.DataFrame({'real': list(y_test_array), 'pred': list(y_pred)}).to_csv(
158     os.path.join(output,
159         f'id={exp_id}_th={use_thickness}_tr={use_traffic}_nh={n_hist}_seed={seed}_linear_regression.csv'))

```

Εικόνα 4.4: Πρόβλεψη δεικτών μέσω της συνάρτησης Linear Regression

Τα αποτελέσματα της πρόβλεψης των δεικτών για το έτος 2021, αποθηκεύονται σε ένα αρχείο .csv, όπου και μπορεί να γίνει η σύγκρισή τους με τους πραγματικούς δείκτες για το αντίστοιχο έτος. Τέλος, το ελάχιστο σφάλμα, αναγράφεται σε ένα

αρχείο μαζί και με τα ελάχιστα σφάλματα των υπόλοιπων μεθόδων, από όπου προκύπτει η ακριβέστερη μέθοδος πρόβλεψης.

4.5.2 Ridge Regression

Όπως αναφέρεται και στο υποκεφάλαιο 3.4.2.3, χαρακτηριστικό της συνάρτησης Ridge Regression είναι η προσθήκη μιας παραμέτρου στην τύπο της μεθόδου των ελαχίστων τετραγώνων. Στον προγραμματισμό, η παράμετρος λ της σχέσης 3.6, ορίζεται με την παράμετρο «alpha». Έτσι, εκτελώντας δοκιμές για τις διάφορες τιμές της «alpha», προκύπτει η βέλτιστη τιμή της («best alpha»). Για τον προσδιορισμό της «best alpha», εκτελούνται 1000 δοκιμές για τιμές της «alpha» από 10^{-12} έως 100.

Με την εισαγωγή της συνάρτησης Ridge Regression (Εικόνα 4.5) και της βέλτιστης τιμής του «alpha», προβλέπονται, μέσω του πολλαπλασιασμού των τελικών δεικτών «x test» με τα κατάλληλα βάρη, οι δείκτες IRI για το έτος 2021 «y test».

```
162 min_err = 10
163 for alpha in np.linspace(1e-12, 100, 1000):
164     reg = Ridge(alpha=alpha).fit(X_train_array, y_train_array)
165     y_pred = reg.predict(X_test_array)
166     error = rmse(y_pred, y_test_array)
167     if error < min_err:
168         min_err = error
169         best_alpha = alpha
170         best_y_pred = y_pred
171         best_reg = reg
172
173 print('Smallest Error for Ridge is:', min_err, 'with alpha:', best_alpha)
174 best_y_pred = list(best_y_pred)
175 _y_test_array = list(y_test_array)
176 for i in range(len(best_y_pred)):
177     best_y_pred[i] = best_y_pred[i]*(_max - _min)+_min
178     _y_test_array[i] = _y_test_array[i]*(_max - _min)+_min
179 pd.DataFrame({'real': list(_y_test_array), 'pred': list(best_y_pred)}).to_csv(
180     os.path.join(output,
181         f'id={exp_id}_th={use_thickness}_tr={use_traffic}_nh={n_hist}_seed={seed}_ridge_regression_alpha={best_alpha}.csv'))
```

Εικόνα 4.5: Πρόβλεψη δεικτών μέσω της συνάρτησης Ridge Regression

Ομοίως, τα αποτελέσματα της πρόβλεψης των δεικτών για το έτος 2021, αποθηκεύονται σε ένα αρχείο .csv, όπου και μπορεί να γίνει η σύγκρισή τους με τους πραγματικούς δείκτες για το αντίστοιχο έτος. Τέλος, το ελάχιστο σφάλμα, αναγράφεται σε ένα αρχείο μαζί και με τα ελάχιστα σφάλματα των υπόλοιπων μεθόδων, από όπου προκύπτει η ακριβέστερη μέθοδος πρόβλεψης.

4.5.3 Lasso Regression

Όπως αναφέρθηκε και στο υποκεφάλαιο 3.4.2.4, χαρακτηριστικό της συνάρτησης Lasso Regression είναι η προσθήκη μιας παραμέτρου στην τύπο της μεθόδου των ελαχίστων τετραγώνων, η οποία διαφέρει με αυτή της Ridge Regression, καθώς η κλίση εισάγεται ως μέγεθος. Η παράμετρος λ της σχέσης 3.7 στο αντίστοιχο κεφάλαιο, εισάγεται ως παράμετρος «alpha». Ακολουθώντας την ίδια διαδικασία με αυτή της συνάρτησης Ridge Regression, προκύπτει έπειτα από δοκιμές η βέλτιστη τιμή «best alpha».

Έτσι, με την εισαγωγή της συνάρτησης Lasso Regression (Εικόνα 4.6) και της βέλτιστης τιμής του «alpha», προβλέπονται, μέσω του πολλαπλασιασμού των τελικών δεικτών «x test» με τα κατάλληλα βάρη, οι δείκτες IRI για το έτος 2021 «y test».

```
184 min_err = 10
185 for alpha in np.linspace(1e-12, 100, 1000):
186     reg = Lasso(alpha=alpha).fit(X_train_array, y_train_array)
187     y_pred = reg.predict(X_test_array)
188     error = rmse(y_pred, y_test_array)
189     if error < min_err:
190         min_err = error
191         best_alpha = alpha
192         best_y_pred = y_pred
193         best_reg = reg
194
195 print('Smallest Error for Lasso is:', min_err, 'with alpha:', best_alpha)
196 best_y_pred = list(best_y_pred)
197 _y_test_array = list(y_test_array)
198 for i in range(len(best_y_pred)):
199     best_y_pred[i] = best_y_pred[i]*(_max - _min)+_min
200     _y_test_array[i] = _y_test_array[i]*(_max - _min)+_min
201 pd.DataFrame({'real': list(_y_test_array), 'pred': list(best_y_pred)}).to_csv(
202     os.path.join(output,
203         f'id={exp_id}_th={use_thickness}_tr={use_traffic}_nh={n_hist}_seed={seed}_lasso_regression_alpha={best_alpha}.csv'))
```

Εικόνα 4.6: Πρόβλεψη δεικτών μέσω της συνάρτησης Lasso Regression

Όμοια με τις προηγούμενες δύο συναρτήσεις, τα αποτελέσματα της πρόβλεψης των δεικτών για το έτος 2021, αποθηκεύονται σε ένα αρχείο .csv, όπου και μπορεί να γίνει η σύγκρισή τους με τους πραγματικούς δείκτες για το αντίστοιχο έτος. Τέλος, το ελάχιστο σφάλμα, αναγράφεται σε ένα αρχείο μαζί και με τα ελάχιστα σφάλματα των υπόλοιπων μεθόδων, από όπου προκύπτει η ακριβέστερη μέθοδος πρόβλεψης.

4.6 Εφαρμογή Νευρωνικών Δικτύων

4.6.1 MLP

Για την λειτουργία του νευρωνικού δικτύου Multilayer Perceptron (υποκεφάλαιο 3.3.2) πραγματοποιούνται, με τη σειρά, οι εξής τρεις βασικές διαδικασίες:

1. Εισαγωγή και διάταξη των δεδομένων στο δίκτυο
2. Παραμετροποίηση της αρχιτεκτονικής του δικτύου
3. Κατασκευή της συνάρτησης εκπαίδευσης

4.6.1.1 Εισαγωγή και διάταξη των δεδομένων στο δίκτυο

Ξεκινώντας με την εισαγωγή των δεδομένων (Dataset), μέσω της συνάρτησης «_init_» (initialize), ορίζονται τα δεδομένα (Εικόνα 4.7). Αναλυτικότερα, μέσω της κλάσης «self» δίνεται η δυνατότητα πρόσβασης στα δεδομένα, τα οποία εισάγονται σε πίνακες (arrays).

```

14 class RoadData(Dataset):
15     def __init__(self, X, y, n_hist, use_thickness, traffic, ds_type='train'):
16         self.n_hist = n_hist
17         self.use_thickness = use_thickness
18         self.X = []
19         self.y = []
20         if self.use_thickness:
21             self._X = X.to_numpy()[ :, 3:]
22             self.thickness = X.to_numpy()[ :, 1:3]
23         else:
24             self._X = X.to_numpy()
25         _array = []
26         n_timepoints = self._X.shape[1]
27         if ds_type == 'train':
28             for i in range(self._X.shape[0]):
29                 if self.use_thickness:
30                     _array_in = np.concatenate([self.thickness[i], self._X[i, :]])
31                 else:
32                     _array_in = self._X[i, :]
33                 self.X.append(_array_in.astype('double'))
34                 self.y.append(y.iloc[i])
35         else:
36             self.X = []
37             self.y = []
38             for i in range(self._X.shape[0]):
39                 if self.use_thickness:
40                     _array_in = np.concatenate([self.thickness[i], self._X[i, :]])
41                 else:
42                     _array_in = self._X[i, :]
43                 self.X.append(_array_in.astype('double'))
44                 self.y.append(y.iloc[i])
45         self.X = np.array(self.X)
46
47         positions = X['km'].tolist()
48         years = [int('20' + y) for y in list(X.keys())[3:]]
49         test_traffic = []
50         for position in positions:
51             test_traffic.append(get_traffic_from_year(position, traffic, years))
52         test_traffic = np.log(np.array(test_traffic))
53         self.X = np.concatenate([self.X, test_traffic], axis=1)
54         self.y = np.array(self.y)

```

Εικόνα 4.7: Εισαγωγή δεδομένων στο MPL (Dataset)

Στη συνέχεια, μέσω της συνάρτησης «_len_» (length), ορίζεται ο αριθμός των δειγμάτων. Τέλος, μέσω της συνάρτησης «_getitem_», δίνεται η δυνατότητα στο δίκτυο να επιλέγει τυχαία δείγματα από τα δεδομένα (Εικόνα 4.8).

```
56     def __len__(self):
57         return self.X.shape[0]
58
59     def __getitem__(self, idx):
60         prev = self.X[idx]
61         pred = self.y[idx]
62         return torch.Tensor(prev), pred
63
```

Εικόνα 4.8: Ορισμός του αριθμού δείγματος και επιλογή τυχαίων δεδομένων

Η επιλογή δειγμάτων με τυχαία σειρά, συμβάλει στην καλύτερη εκπαίδευση του δικτύου.

4.6.1.2 Παραμετροποίηση της αρχιτεκτονικής του δικτύου

Για την παραμετροποίηση της αρχιτεκτονικής του δικτύου ορίζονται 2 επίπεδα, από τα οποία καλούνται να περάσουν τα στοιχεία (Εικόνα 4.9). Από το πρώτο επίπεδο (input layer) διέρχονται όλα τα στοιχεία δεδομένων, δηλαδή τα έτη, τα πάχη και τα στοιχεία κυκλοφορίας (συνολικά 18 στοιχεία). Τα στοιχεία αυτά, εξερχόμενα από το πρώτο επίπεδο, μικραίνουντας σε αριθμό και στη συνέχεια διέρχονται από μία μη γραμμική συνάρτηση ReLU, όπου τα στοιχεία αλλάζουν σε τιμή, αλλά όχι σε αριθμό. Τέλος, τα στοιχεία διέρχονται από το δεύτερο επίπεδο (output layer), όπου και καταλήγουν σε ένα μοναδικό στοιχείο.



Εικόνα 4.9: Επίπεδα αρχιτεκτονικής MLP

Τα δύο επίπεδα, αποτελούνται από μία απλή γραμμική συνάρτηση (linear) (Εικόνα 4.10). Η μη γραμμική συνάρτηση, που περιβάλλεται των δύο επιπέδων, πρόκειται για τη συνάρτηση ενεργοποίησης Relu (REctified Linear Units).

```

68 class MLP(nn.Module):
69     def __init__(self, num_features):
70         super().__init__()
71         self.x1 = nn.Linear(num_features, 8)
72         self.relu = nn.ReLU()
73         self.x2 = nn.Linear(8, 1)
74
75     def forward(self, x):
76         x = self.x1(x)
77         x = self.relu(x)
78         x = self.x2(x)
79         return x
80

```

Εικόνα 4.10: Αρχιτεκτονική MLP

Με την εισαγωγή μιας συνάρτησης ενεργοποίησης, μεταξύ δύο γραμμικών επιπέδων, επιτυγχάνεται μία μη γραμμικότητα στο δίκτυο, προσφέροντας στο δίκτυο την ικανότητα να βρει μοτίβα, τα οποία δε θα μπορούσαν να προκύψουν από μία απλή γραμμική συνάρτηση.

4.6.1.3 Κατασκευή της συνάρτησης εκπαίδευσης

Με την έναρξη της διαδικασίας της εκπαίδευσης, εισάγεται ένα «seed», με το οποίο επιτυγχάνεται η σωστή αρχικοποίηση των βαρών. Πιο συγκεκριμένα, τα βάρη ξεκινάνε στην αρχή της εκπαίδευσης με μια τυχαία τιμή. Με την εισαγωγή ενός «seed», επιτυγχάνεται η επανάληψη της διαδικασίας του πειράματος με τα ίδια ακριβώς αποτελέσματα, καθώς πολλές διαδικασίες γίνονται με στοχαστικό τρόπο.

Για τις διάφορες χιλιομετρικές θέσεις του κάθε τμήματος της οδού, επιλέγεται το 80% των θέσεων για τη χρήση τους ως δεδομένα εκπαίδευσης του δικτύου. Το υπόλοιπο 20% των χιλιομετρικών θέσεων, χρησιμοποιείται για την πρόβλεψη των μελλοντικών δεικτών (test dataset). Όταν το δίκτυο εκπαιδευτεί επαρκώς, η επαναληπτική διαδικασία της εκπαίδευσης σταματάει. Η αναλυτική διαδικασία διαχωρισμού των δεδομένων σε δεδομένα εκπαίδευσης και τεστ, παρουσιάζεται στην Εικόνα 4.11.

```

82 def train_mlp(seed, experiment, X_train, y_train, X_test, y_test, traffic, use_traffic, use_thickness, n_hist, min, max):
83     if not os.path.exists('models'):
84         os.makedirs('models')
85     torch.manual_seed(seed)
86     local_break_idx = int(len(X_train) * 0.8)
87     train_dataset = RoadData(X_train.iloc[:local_break_idx], y_train.iloc[:local_break_idx], n_hist, use_thickness,
88                             traffic, ds_type='train') # 0.8 * 0.8 = 64%
89     trainloader = torch.utils.data.DataLoader(train_dataset, batch_size=256, shuffle=True)
90
91     dev_dataset = RoadData(X_train.iloc[local_break_idx:], y_train.iloc[local_break_idx:], n_hist, use_thickness,
92                             traffic, ds_type='dev') # 0.8 * 0.2 = 16%
93     devloader = torch.utils.data.DataLoader(dev_dataset, batch_size=256, shuffle=False)
94
95     test_dataset = RoadData(X_test, y_test, n_hist, use_thickness, traffic, ds_type='test') # 0.2 = 20%
96     testloader = torch.utils.data.DataLoader(test_dataset, batch_size=256, shuffle=False)

```

Εικόνα 4.11: Διαχωρισμών train dataset και test dataset στο MLP

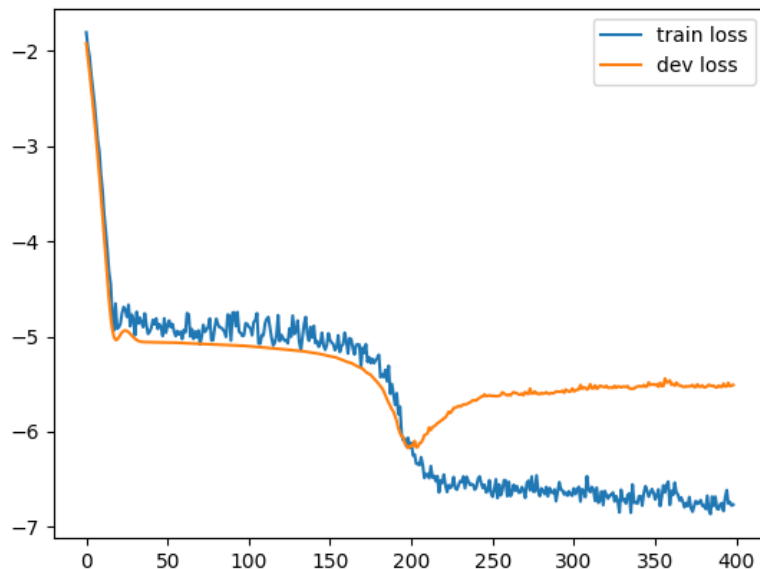
Για την εκπαίδευση, ορίζεται ως «train dataset» (σύνολο δεδομένων εκπαίδευσης) το 80% των δεδομένων της εκπαίδευσης, δηλαδή το 64% των συνολικών δεδομένων εισόδου. Επίσης, ως «development dataset» (σύνολο δεδομένων ανάπτυξης) ορίζεται το 20% των δεδομένων της εκπαίδευσης, δηλαδή το 16% των συνολικών δεδομένων εισόδου.

Στη διαδικασία της εκπαίδευσης, μέσω των δεδομένων του «train dataset», πολλαπλασιάζονται τα δεδομένα εισόδου «x train», για διάφορες χιλιομετρικές θέσεις, με τα κατάλληλα βάρη, δίνοντας κάθε φορά την εκτίμηση ενός δείκτη IRI. Στη συνέχεια, και σε ομάδες (batches) των 256, πραγματοποιείται μία πίσω διάδοση (back propagation), όπου τροποποιούνται τα βάρη σύμφωνα με το μέσο όρο τους, επαληθεύοντας τις εκτιμήσεις με τους πραγματικούς δείκτες IRI («y train»).

Παράλληλα με τη διαδικασία της εκπαίδευσης, μέσω του συνόλου των στοιχείων του «development dataset», πραγματοποιείται ο έλεγχος επαρκούς εκπαίδευσης του δικτύου. Πιο συγκεκριμένα, μέσω των δεδομένων του «development dataset», πολλαπλασιάζονται τα δεδομένα εισόδου «x train», για διάφορες χιλιομετρικές θέσεις, με τα κατάλληλα βάρη, δίνοντας κάθε φορά την εκτίμηση ενός δείκτη IRI. Η διαδικασία διαφέρει από αυτή της εκπαίδευσης, καθώς τώρα δεν πραγματοποιείται διόρθωση των βαρών.

Με τη σύγκριση των προβλέψεων των δεδομένων του «development dataset» με τους πραγματικούς δείκτες, προκύπτει ένα σφάλμα (development loss), το οποίο συγκρίνεται με σφάλμα το της εκπαίδευσης (train loss). Η διαδικασία της εκπαίδευσης του δικτύου διακόπτεται όταν παρατηρηθεί μείωση του σφάλματος εκπαίδευσης, χωρίς τη μείωση του σφάλματος ανάπτυξης. Με την «υπερβολική» εκπαίδευση του δικτύου, γίνεται υπερπροσαρμογή του δικτύου στα δεδομένα εκπαίδευσης (overfitting), με αποτέλεσμα το δίκτυο να παρέχει καλά αποτελέσματα στην εκπαίδευση, αλλά όχι τόσο καλά αποτελέσματα στις τελικές προβλέψεις.

Στο Σχήμα 4.5, απεικονίζεται η διαφορά σφαλμάτων εκπαίδευσης (training) και ανάπτυξης (development).



Σχήμα 4.5: Διαφορά σφαλμάτων train και development

Με την εισαγωγή του νευρωνικού δικτύου MLP (Εικόνα 4.12), καθορίζεται η συνάρτηση σφάλματος, η οποία δεν είναι άλλη από τη Μέση Τετραγωνική Απώλεια Σφάλματος (Mean Squared Error Loss). Επίσης καθορίζεται και η συνάρτηση βελτιστοποίησης του σφάλματος (optimizer). Με τη συνάρτηση «optimizer» επιτυγχάνεται η ελάττωση του σφάλματος, βρίσκοντας το ελάχιστο δυνατό σφάλμα.

```
98      # Initialize the MLP
99      mlp = MLP(num_features= 8 + 2 * use_thickness + 8 * 2)
100     mlp = mlp.double()
101
102     # Define the loss function and optimizer
103     loss_function = nn.MSELoss()
104     optimizer = torch.optim.SGD(mlp.parameters(), lr=1e-3)
105     optimizer = torch.optim.Adam(mlp.parameters())
```

Εικόνα 4.12: Εισαγωγή MLP και καθορισμός της συνάρτησης σφάλματος

Έπειτα, ορίζονται οι επαναλήψεις, καλούμενες ως «εποχές» για το δίκτυο (Εικόνα 4.13). Σε κάθε εποχή, πραγματοποιείται μία επανάληψη σε όλα τα δεδομένα. Δεδομένου ότι όσοι περισσότερες επαναλήψεις κάνει το δίκτυο, τόσο καλύτερα

μαθαίνει και τόσο μικρότερο το σφάλμα, έστω ότι ο αριθμός των εποχών είναι 1000.

Στη συνέχεια, αφού ορίζονται τα δεδομένα και οι στόχοι του δικτύου, μηδενίζεται ο βελτιστοποιητής (optimizer), καθώς με κάθε επανάληψη συσσωρεύει παραγώγους. Στο τέλος κάθε επανάληψης γίνεται μία πίσω διάδοση (back propagation) τροποποιώντας τα βάρη («optimizer.step»).

Μέσω της συνάρτησης «loss», υπολογίζεται η απώλεια του δικτύου, συγκρίνοντας τις πραγματικές μετρήσεις με τις προβλεπόμενες από το δίκτυο. Με την εντολή «loss backward», υπολογίζεται ο αριθμός των παραγώγων που πρέπει να αλλάξουν, οι οποίες αλλάζουν με την εντολή «optimizer.step», όπως προαναφέρθηκε. Τελευταίο βήμα, αυτής της επαναληπτικής διαδικασίας, είναι ο υπολογισμός του σφάλματος (current loss), μετά από κάθε επανάληψη.

```
108     es_best_score = 10000
109     es_best_epoch = -1
110     train_loss = []
111     dev_loss = []
112     wait = 0
113     es_triggered = False
114     for epoch in tqdm(range(0, 10000)):
115         mlp.train()
116         current_loss = 0.0
117         for i, data in enumerate(trainloader, 0):
118             inputs, targets = data
119             optimizer.zero_grad()
120             outputs = mlp(inputs.double())
121             loss = loss_function(outputs.flatten(), targets)
122             loss.backward()
123             grads = []
124             optimizer.step()
125             current_loss += loss.item()
126         current_loss /= len(trainloader)
127         train_loss.append(current_loss)
```

Εικόνα 4.13: Καθορισμός των επαναλήψεων και υπολογισμός του σφάλματος σε κάθε επανάληψη στο MLP

Για να πραγματοποιηθεί σωστά ο έλεγχος, εισάγεται η εντολή «with torch no grad», ώστε να μη συσσωρευτούν τα βάρη, καθώς σε αυτό το βήμα έχει τελειώσει η εκπαίδευση. Επίσης εισάγεται η εντολή «mlp.eval», μέσω της οποίας σβήνονται οι στοχαστικότητατες που λειτουργούν κατά την εκπαίδευση, ώστε να πραγματοποιείται η επαλήθευση με τον ίδιο τρόπο κάθε φορά (Εικόνα 4.14).

Για τη διακοπή της εκπαίδευσης, ορίζεται ένα μέγιστο εύρος επαναλήψεων (έστω 10.000), σύμφωνα με το οποίο, αν το σφάλμα της εξέλιξης σταματήσει να μειώνεται

αναλογικά με το σφάλμα της εκπαίδευσης, η εκπαίδευση σταματάει έπειτα από 200 επαναλήψεις.

```
131     for i, data in enumerate(devloader, 0):
132         inputs, targets = data
133         mlp.eval()
134         with torch.no_grad():
135             outputs = mlp(inputs.double())
136             loss = loss_function(outputs.flatten(), targets)
137             current_loss += loss.item()
138         current_loss /= len(devloader)
139         dev_loss.append(current_loss)
140         if current_loss < es_best_score:
141             es_best_score = current_loss
142             es_best_epoch = epoch
143             es_best_model = mlp
144             wait = 0
145         else:
146             wait += 1
147
148         if wait == 200:
149             print('ES triggered. Best epoch is epoch: {}'.format(es_best_epoch))
150             es_triggered = True
151             break
152     print('Training process has finished.')
```

Εικόνα 4.14: Επαλήθευση δεδομένων και διακοπή εκπαίδευσης MLP

Όπως απεικονίζεται στο διάγραμμα του Σχήματος 4.5, το development loss σταματάει να μειώνεται αναλογικά με το train loss μετά από 200 επαναλήψεις. Έτσι, με την ολοκλήρωση άλλων 200 επαναλήψεων, όπως ορίσθηκε, η εκπαίδευση ολοκληρώνεται μετά από 400 επαναλήψεις.

Με την ολοκλήρωση της διαδικασίας, φορτώνεται το βέλτιστο μοντέλο εκπαίδευσης και με τη βοήθεια των στοιχείων του «test dataset» γίνεται η πρόβλεψη των μελλοντικών δεικτών IRI για το έτος 2021, υπολογίζοντας το σφάλμα του νευρωνικού δικτύου (Εικόνα 4.15).

```

154     test_loss = 0
155     if es_triggered:
156         pass
157     es_best_model.eval()
158     predicted = []
159     real = []
160     with torch.no_grad():
161         for i, data in enumerate(testloader, 0):
162             inputs, targets = data
163             outputs = es_best_model(inputs.double())
164             real += targets.tolist()
165             predicted += [l[0] for l in outputs.tolist()]
166             loss = rmse(outputs.flatten(), targets)
167             test_loss += loss.item()
168     test_loss /= len(testloader)
169     filename = f'results/id={experiment}_th={use_thickness}_tr={use_traffic}_nh={n_hist}_seed={seed}_mlp.csv'
170     for i in range(len(real)):
171         real[i] = real[i]*(max-min)+min
172         predicted[i] = predicted[i]*(max-min)+min
173     pd.DataFrame({'real': real, 'predicted': predicted}).to_csv(filename)
174     print('MLP test loss:', test_loss)
175
176     plt.cla()
177     plt.plot(train_loss, label='train loss')
178     plt.plot(dev_loss, label='dev loss')
179     plt.legend()
180     plt.savefig(f'figs/mlp_train_loss_{n_hist}_{experiment}.png')
181
182     plt.cla()
183     plt.plot([np.log(p) for p in train_loss], label='train loss')
184     plt.plot([np.log(p) for p in dev_loss], label='dev loss')
185     plt.legend()
186     plt.savefig(f'figs/mlp_train_loss_{n_hist}_{experiment}_log.png')

```

Εικόνα 4.15: Τέλος εκπαίδευσης και υπολογισμός βέλτιστου μοντέλου του Νευρωνικού Δικτύου MLP

Τέλος, τα αποτελέσματα της πρόβλεψης των δεικτών IRI για το έτος 2021, αποθηκεύονται σε ένα αρχείο .csv. Το σφάλμα του δικτύου, αναγράφεται σε ένα αρχείο μαζί και με τα ελάχιστα σφάλματα των υπόλοιπων μεθόδων, από όπου προκύπτει η ακριβέστερη μέθοδος πρόβλεψης. Επιπλέον, δημιουργούνται διαγράμματα απεικονίζοντας το σφάλμα της εκπαίδευσης (πχ Σχήματα 4.5).

Σημειώνεται ότι, τα δεδομένα του «test dataset» επιλέγονται τυχαία και αποτελούν το 20% των χιλιομετρικών θέσεων κάθε τμήματος της οδού. Επομένως, στα διαγράμματα των προβλέψεων, που παραθέτονται στο Κεφάλαιο 5, απεικονίζεται το σφάλμα μεταξύ των προβλέψεων, για αυτές τις τυχαίες θέσεις, και των πραγματικών τιμών τους, και όχι οι δείκτες IRI κατά μήκος της οδού.

4.6.2 LSTM

Ως δεύτερο Νευρωνικό Δίκτυο επιλέγεται το δίκτυο LSTM (Long Short Term Memory) (υποκεφάλαιο 3.3.4), καθώς με τη χρονική αντίληψή του, αλλά και την ικανότητά του να αποθηκεύει δεδομένα για μεγάλο χρονικό διάστημα, αποτελεί μία αναξιόπιστη μέθοδο για μελλοντικές προβλέψεις.

Για την κατασκευή του δικτύου LSTM, ακολουθούνται τα ίδια βήματα με αυτά της κατασκευής του δικτύου MLP, με κάποιες βασικές διαφορές. Παρακάτω παρουσιάζεται αναλυτικά η διαδικασία.

4.6.2.1 Εισαγωγή και διάταξη των δεδομένων στο δίκτυο

Πρώτο βήμα της κατασκευής του δικτύου LSTM (Long Short Term Memory), είναι η εισαγωγή των δεδομένων στο δίκτυο με την εντολή «_init_» (initialize) και η δυνατότητα πρόσβασης σε αυτά, μέσω της παραμέτρου «self».

Όπως και προηγουμένως, μέσω της εντολής «_len_» (length), ορίζεται ο αριθμός των δειγμάτων και με την εντολή «_getitem_», δίνεται η δυνατότητα στο δίκτυο να επιλέγει τυχαία δείγματα από τα δεδομένα (Εικόνα 4.16).

```
15 class RoadData(Dataset):
16     def __init__(self, X, y, traffic, use_thickness=True):
17         self.X = X
18         self.y = y
19         self.traffic = traffic
20         self.use_traffic = True
21         self.use_thickness = True
22
23     def __len__(self):
24         return self.X.shape[0]
25
26     def __getitem__(self, idx):
27         years = [int('20' + y) for y in find_index_columns(self.X)[0]]
28         X = self.X.iloc[idx][find_index_columns(self.X)[0]].to_numpy().astype(float)
29         km = float(self.X['km'].tolist()[idx].split('_')[0])
30         lane = int(self.X['km'].tolist()[idx].split('_')[1][0])
31         position = self.X['km'].tolist()[idx].split('_')[1][1]
32
33         traffic = []
34         if self.use_traffic:
35             for km_pair in self.traffic.keys():
36                 if float(km_pair.split('-')[0]) <= km <= float(km_pair.split('-')[1]):
37                     traffic = []
38                     for year in years:
39                         traffic.append(List((self.traffic[km_pair][year][position + 'L'])))
40
41         thickness = []
42         if self.use_thickness:
43             heights = self.X.iloc[idx]
44             thickness = [heights['h1'], heights['h2']]
45
46         pred = self.y.iloc[idx]
47         return torch.tensor(X), torch.log(torch.tensor(traffic)), torch.tensor(thickness), torch.tensor(pred)
```

Εικόνα 4.16: Εισαγωγή και διάταξη δεδομένων

Με τη συνάρτηση «years» επιλέγεται η στήλη του έτους για την οποία χρησιμοποιούνται τα αντίστοιχα δεδομένα. Με το δείκτη «y» εισάγονται τα δύο τελευταία ψηφία του έτους. Για παράδειγμα, για άντληση δεδομένων από τη στήλη του έτους 2020, ο δείκτης θα είναι ίσος με «20».

Ακόμα, με το δείκτη «X» συμβολίζονται όλα τα δεδομένα εισόδου, τα οποία εκτός από τους δείκτες IRI, περιέχουν και τα πάχη των δύο τμημάτων. Έτσι, με την

συνάρτηση «find index columns», παραμένουν μόνο τα στοιχεία που αντιστοιχούν στη συνάρτηση «years», άρα τους δείκτες IRI, οι οποίοι διαχωρίζονται ανάλογα με το τμήμα στο οποίο ανήκουν, τη λωρίδα και το ίχνος.

Στη συνέχεια, πραγματοποιείται ο διαχωρισμός των δεδομένων της κίνησης (traffic), της κάθε χιλιομετρικής θέσης, στο αντίστοιχο τμήμα (Α ή Β), και η εισαγωγή των δεδομένων σε πίνακα. Τέλος, εισάγονται τα πάχη του οδοστρώματος.

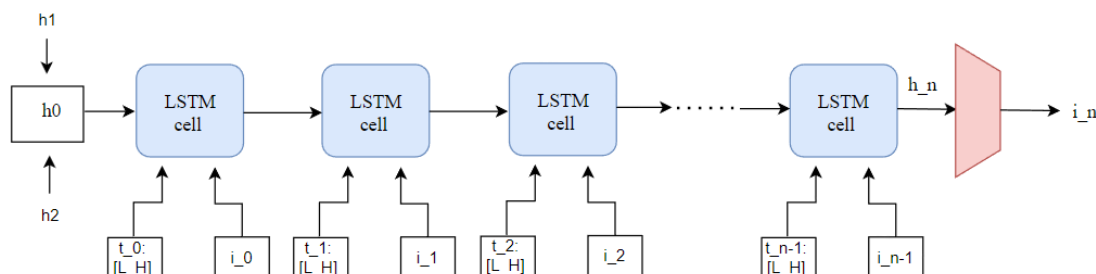
4.6.2.2 Παραμετροποίηση της αρχιτεκτονικής του δικτύου

Τα Νευρωνικά Δίκτυα LSTM, χαρακτηρίζονται από τα κρυφά επίπεδα, μέσα στα οποία πραγματοποιείται μία σειρά αριθμητικών πράξεων. Έτσι, ορίζεται ο αριθμός των επιπέδων του δικτύου (2) και ο αριθμός των κρυφών επιπέδων (16). Στην έξοδο του δικτύου, ορίζεται ένα γραμμικό επίπεδο (linear), ώστε να εξάγεται ένα και μοναδικό στοιχείο (Εικόνα 4.17).

```
50 class ShallowRegressionLSTM(nn.Module):
51     def __init__(self, num_sensors, hidden_units, use_traffic=True, use_thickness=True):
52         super().__init__()
53         self.num_sensors = num_sensors
54         self.hidden_units = hidden_units
55         self.num_layers = 2
56
57         self.use_traffic = use_traffic
58         self.use_thickness = use_thickness
59         self.lstm = nn.LSTM(
60             input_size=num_sensors,
61             hidden_size=hidden_units,
62             batch_first=True,
63             num_layers=self.num_layers
64         )
65
66         self.linear = nn.Linear(in_features=self.hidden_units, out_features=1)
67
68     def forward(self, x, traffic, thickness):
69         x = x[:, :, None]
70         if self.use_traffic:
71             x = torch.concatenate([x, traffic], dim=2)
72         batch_size = x.shape[0]
73         if self.use_thickness:
74             h0 = torch.zeros(self.num_layers, batch_size, self.hidden_units)# .requires_grad_()
75             for ibatch in range(h0.shape[1]):
76                 h0[:, ibatch, :8] = thickness[ibatch, 0]
77                 h0[:, ibatch, 8:] = thickness[ibatch, 1]
78             h0 = h0.requires_grad_()
79         else:
80             h0 = torch.zeros(self.num_layers, batch_size, self.hidden_units).requires_grad_()
81         c0 = torch.zeros(self.num_layers, batch_size, self.hidden_units).requires_grad_()
82         _, (hn, _) = self.lstm(x, (h0, c0))
83         out = self.linear(hn[0]).flatten()
84         return out
```

Εικόνα 4.17: Εισαγωγή των επιπέδων του δικτύου LSTM

Ορίζεται ένας δυσδιάστατος χρονικός πίνακας «x», στον οποίο εισάγονται, για κάθε χιλιομετρική θέση, οι δείκτες IRI και η κίνηση. Έπειτα, εισάγονται στο πρώτο κρυφό επίπεδο τα πάχη της ασφαλικής στρώσης και τα πάχη της βάσης-υπόβασης, για κάθε χιλιομετρική θέση. Έτσι, δημιουργείται το δίκτυο της μορφής της εικόνας 4.18.



Εικόνα 4.18: Δομή του δικτύου LSTM

Όπως απεικονίζεται στην Εικόνα 4.18, στο πρώτο κρυφό επίπεδο εισάγονται τα πάχη του οδοστρώματος, καθώς αυτά δεν αλλάζουν χρονικά. Εν συνεχεία, στα υπόλοιπα κρυφά επίπεδα, εισάγονται τα στοιχεία που μεταβάλλονται χρονικά ανά έτος, δηλαδή οι δείκτες IRI και τα στοιχεία κυκλοφορίας. Τέλος, τα στοιχεία διέρχονται από ένα γραμμικό επίπεδο, όπου καταλήγουν σε ένα μοναδικό στοιχείο πρόβλεψης.

4.6.2.3 Κατασκευή της συνάρτησης εκπαίδευσης

Η διαδικασία κατασκευής της συνάρτησης εκπαίδευσης του δικτύου LSTM, είναι ίδια με αυτή του δικτύου MLP. Πιο αναλυτικά και όμοια με προηγουμένως, εισάγεται ένα «seed», για τη σωστή αρχικοποίηση των βαρών. Έπειτα, χωρίζονται τα δεδομένα σε ποσοστά 80-20%, όπου το 80% των δεδομένων αξιοποιείται για την εκπαίδευση του δικτύου, ενώ με το 20% πραγματοποιούνται οι μελλοντικές προβλέψεις, μετά το πέρας της εκπαίδευσης (Εικόνα 4.19).

Επίσης, τα δεδομένα εκπαίδευσης (80% εκ του συνόλου) χωρίστηκαν σε δύο κατηγορίες, με ποσοστά 80-20%. Με την πρώτη κατηγορία («train dataset»), επιτυγχάνεται η εκπαίδευση του δικτύου, ενώ με τη δεύτερη («development dataset»), ελέγχεται η επαρκής εκπαίδευση του δικτύου. Όταν το δίκτυο εκπαιδευτεί επαρκώς, η επαναληπτική διαδικασία διακόπτεται. Η πιο αναλυτική λειτουργία των δύο κατηγοριών έχει δοθεί στο υποκεφάλαιο 4.6.1.3.

```

87 def train_lstm(seed, experiment, X_train, y_train, X_test, y_test, traffic, use_traffic, use_thickness, n_hist, min, max):
88     if not os.path.exists('models'):
89         os.makedirs('models')
90     torch.manual_seed(seed)
91     local_break_idx = int(len(X_train) * 0.8)
92     train_dataset = RoadData(X_train.iloc[:local_break_idx], y_train.iloc[:local_break_idx], traffic, n_hist, use_thickness)
93     trainloader = torch.utils.data.DataLoader(train_dataset, batch_size=256, shuffle=True)
94
95     dev_dataset = RoadData(X_train.iloc[local_break_idx:], y_train.iloc[local_break_idx:], traffic)
96     devloader = torch.utils.data.DataLoader(dev_dataset, batch_size=4, shuffle=False)
97
98     test_dataset = RoadData(X_test, y_test, traffic)
99     testloader = torch.utils.data.DataLoader(test_dataset, batch_size=4, shuffle=False)
100
101     num_hidden_units = 16
102
103     use_traffic = True
104     if use_traffic:
105         num_sensors = 3
106     else:
107         num_sensors = 1

```

Εικόνα 4.19: Διαχωρισμών train dataset και test dataset στο LSTM

Σε επόμενο βήμα, εισάγεται το μοντέλο του Νευρωνικού Δικτύου LSTM, ορίζεται η συνάρτηση σφάλματος (Mean Squared Error Loss) και ορίζεται η συνάρτηση βελτιστοποίησης του σφάλματος (optimizer)(Εικόνα 4.20). Για καλύτερα και πιο γρήγορα αποτελέσματα, χρησιμοποιείται η συνάρτηση optimizer Adam.

```

109 model = ShallowRegressionLSTM(num_sensors=num_sensors, use_traffic=use_traffic, hidden_units=num_hidden_units)
110 loss_function = nn.MSELoss()
111 optimizer = torch.optim.Adam(model.parameters())

```

Εικόνα 4.20: Εισαγωγή LSTM και καθορισμός της συνάρτησης σφάλματος

Έπειτα, ορίζονται οι εποχές του δικτύου (έστω 1000) και μηδενίζεται η συσσώρευση σφάλματος του βελτιστοποιητή (optimizer). Στο τέλος κάθε επανάληψης γίνεται μία πίσω διάδοση (back propagation), όπου τροποποιούνται τα βάρη («optimizer.step»).

Μέσω της συνάρτησης «loss», υπολογίζεται η απώλεια του δικτύου, συγκρίνοντας τις πραγματικές μετρήσεις με τις προβλεπόμενες από το δίκτυο. Με την εντολή «loss backward», υπολογίζεται ο αριθμός των παραγώγων που πρέπει να αλλάξουν, οι οποίες αλλάζουν με την εντολή «optimizer.step», όπως προαναφέρθηκε. Τέλος, υπολογίζεται το σφάλμα (current loss), μετά από κάθε επανάληψη (Εικόνα 4.21).

```

114     es_best_score = 100
115     es_best_epoch = -1
116     train_loss = []
117     dev_loss = []
118     wait = 0
119     for epoch in tqdm(range(0, 1000)):
120         model.train()
121         current_loss = 0.0
122         for i, data in enumerate(trainloader, 0):
123             X, traffic_data, thickness_data, pred = data
124             optimizer.zero_grad()
125             outputs = model(X.float(), traffic_data, thickness_data)
126             loss = loss_function(outputs, pred.float())
127             loss.backward()
128             optimizer.step()
129             current_loss += loss.item()
130         current_loss /= len(trainloader)
131         train_loss.append(current_loss)
132         if epoch % 10 == 0:
133             print('Epoch: ', epoch)
134             print('Train Loss: %.3f' %
135                   (current_loss))

```

Εικόνα 4.21: Καθορισμός των επαναλήψεων και υπολογισμός του σφάλματος σε κάθε επανάληψη στο LSTM

Στο τέλος κάθε επανάληψης, με την εντολή «with torch no grad», καθώς και την εντολή «mlp.eval», σβήνονται τα βάρη και οι στοχαστικότητες, για τη σύγκριση των ιδίων χιλιομετρικών θέσεων (Εικόνα 4.22).

Ακόμα, ορίζεται ένα εύρος επαναλήψεων (έστω 200), σύμφωνα με το οποίο, αν το σφάλμα της εξέλιξης σταματήσει να μειώνεται αναλογικά με το σφάλμα της εκπαίδευσης, η εκπαίδευση σταματάει έπειτα από 200 επαναλήψεις.

```

139         for i, data in enumerate(devloader, 0):
140             X, traffic_data, thicknes_data, pred = data
141             model.eval()
142             with torch.no_grad():
143                 outputs = model(X.float(), traffic_data, thicknes_data)
144                 loss = loss_function(outputs, pred.float())
145
146                 current_loss += loss.item()
147             current_loss /= len(devloader)
148
149             dev_loss.append(current_loss)
150             if epoch % 10 == 0:
151                 print('dev Loss: %.3f' %
152                       (current_loss))
153
154             if current_loss < es_best_score:
155                 es_best_score = current_loss
156                 es_best_epoch = epoch
157                 es_best_model = model
158                 wait = 0
159             else:
160                 wait += 1
161
162             if wait == 200:
163                 print('ES triggered. Best epoch is epoch: {}'.format(es_best_epoch))

```

Εικόνα 4.22: Επαλήθευση δεδομένων και διακοπή εκπαίδευσης LSTM

Τα αποτελέσματα της πρόβλεψης των δεικτών IRI για το έτος 2021, αποθηκεύονται σε ένα αρχείο .csv. Το σφάλμα του δικτύου αναγράφεται σε ένα αρχείο, μαζί και με τα ελάχιστα σφάλματα των υπόλοιπων μεθόδων, από όπου προκύπτει η ακριβέστερη μέθοδος πρόβλεψης, ακόμα δημιουργούνται διαγράμματα απεικονίζοντας το σφάλμα της εκπαίδευσης (Εικόνα 4.23).

```

168     test_loss = 0
169     predicted = []
170     real = []
171     for i, data in enumerate(testloader, 0):
172         X, traffic_data, thicknes_data, pred = data
173         model.eval()
174         with torch.no_grad():
175             outputs = model(X.float(), traffic_data, thicknes_data)
176             predicted += outputs.tolist()
177             real += pred.tolist()
178             loss = loss_function(outputs, pred.float())
179             test_loss += loss.item()
180     filename = f'results/id={experiment}_th={use_thickness}_tr={use_traffic}_nh={n_hist}_seed={seed}_lstm.csv'
181     for i in range(len(real)):
182         real[i] = real[i]*(max-min)+min
183         predicted[i] = predicted[i]*(max-min)+min
184     pd.DataFrame({'real': real, 'predicted': predicted}).to_csv(filename)
185     test_loss /= len(devloader)
186     print('LSTM test Loss: {:.3f}'.format(current_loss))
187
188     plt.cla()
189     plt.plot(train_loss, label='train loss')
190     plt.plot(dev_loss, label='dev loss')
191     plt.legend()
192     plt.savefig(f'figs/lstm_train_loss_{n_hist}_{experiment}.png')

```

Εικόνα 4.23: Τέλος εκπαίδευσης και υπολογισμός βέλτιστου μοντέλου του Νευρωνικού Δικτύου LSTM

Με την εκτέλεση όλων των προαναφερθέντων μεθόδων, προκύπτει το σφάλμα κάθε μίας για κάθε πείραμα, όπως παραθέτεται στο Κεφάλαιο 5 των αποτελεσμάτων. Έπειτα, με τη σύγκριση των σφαλμάτων, προκύπτει η βέλτιστη μέθοδος για την εκτίμηση της μελλοντικής ομαλότητας του οδοστρώματος μίας οδού.

5. Αποτελέσματα

5.1 Αποτελέσματα 1^{ης} πειραματικής ανάλυσης

Το σφάλμα κάθε μεθόδου, για την εκτέλεση της 1^{ης} πειραματικής ανάλυσης, απεικονίζεται στην Εικόνα 5.1.

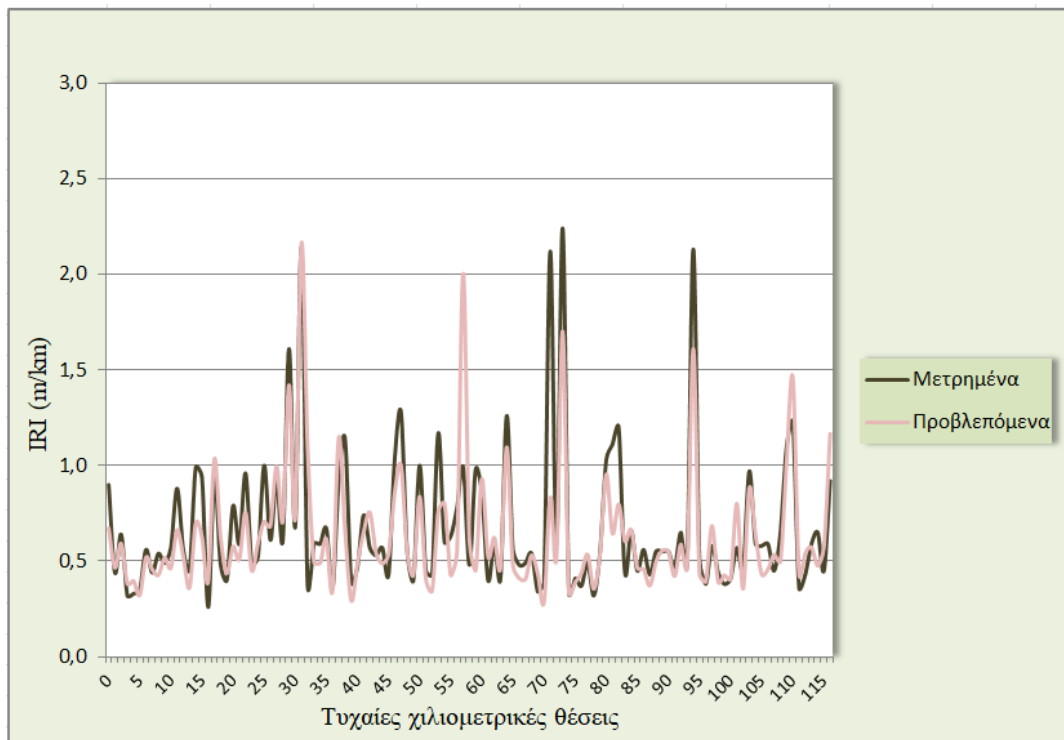
```
Error for LinearRegression is: 0.02682008584421341  
Smallest Error for Ridge is: 0.025537907177115866 with alpha: 0.7007007007016938  
Smallest Error for Lasso is: 0.026820085820490794 with alpha: 1e-12  
MLP test loss: 0.026307847696442715  
LSTM test Loss: 0.001
```

Εικόνα 5.1: Σφάλματα μεθόδων πρόβλεψης για την 1^η πειραματική ανάλυση

Σύμφωνα με τα αποτελέσματα της εικόνας 5.1, παρατηρείται ότι τα Νευρωνικά Δίκτυα MLP και LSTM, αποδίδουν καλύτερα στην εκτίμηση των μελλοντικών δεικτών IRI, συγκριτικά με τις γραμμικές συναρτήσεις.

Αναφορικά με τις γραμμικές μεθόδους πρόβλεψης, καλύτερη μέθοδος, όπως και αναμενόταν, αναδείχθηκε η συνάρτηση Ridge Regression, καθώς ως γνωστόν, αποδίδει καλύτερα όταν τα δεδομένα εισόδου (στη συγκεκριμένη περίπτωση τα πάχη του οδοστρώματος και τα στοιχεία κυκλοφορίας) σχετίζονται άμεσα με την πρόβλεψη των δεικτών IRI. Σε αντίθετη περίπτωση, θα αναμενόταν μικρότερο σφάλμα από τη γραμμική συνάρτηση Lasso Regression.

Ως η βέλτιστη μέθοδος πρόβλεψης της 1^{ης} πειραματικής ανάλυσης, αναδείχθηκε αυτή του Νευρωνικού Δικτύου LSTM, της οποίας τα αποτελέσματα απεικονίζονται στο διάγραμμα του σχήματος 5.1.



Σχήμα 5.1: Πρόβλεψη των δεικτών IRI της αριστερής λωρίδας του τμήματος Α και σύγκριση με τους πραγματικούς δείκτες για το ίδιο έτος – 1^η πειραματική ανάλυση

Σύμφωνα και με το Σχήμα 5.1, οι προβλέψεις του δικτύου δεν αποκλίνουν σημαντικά από τις πραγματικές τιμές των δεικτών, κάτι το οποίο καθιστά δυνατή την πρόβλεψη της ομαλότητας του οδοστρώματος μίας οδού, μέσω ενός δικτύου LSTM.

5.2 Αποτελέσματα 2^{ης} πειραματικής ανάλυσης

Το σφάλμα κάθε μεθόδου, για την εκτέλεση της 2^{ης} πειραματικής ανάλυσης, απεικονίζεται στην Εικόνα 5.2.

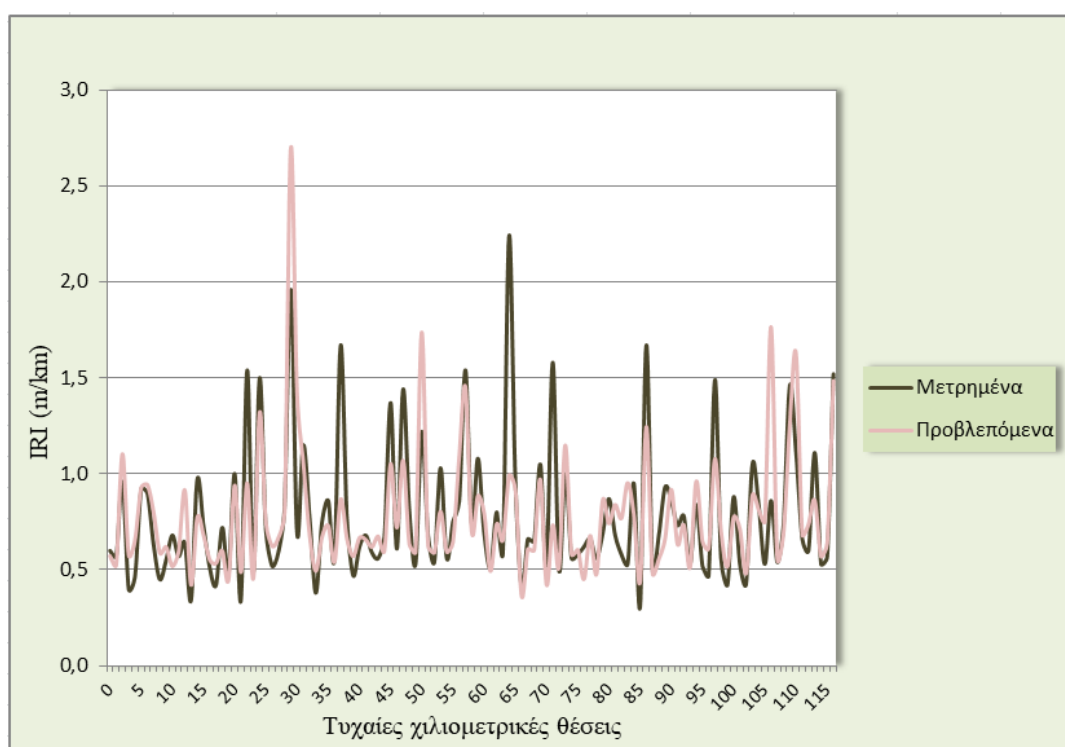
```
Error for LinearRegression is: 0.03403887544400121
Smallest Error for Ridge is: 0.03403887544400322 with alpha: 1e-12
Smallest Error for Lasso is: 0.034038875458195755 with alpha: 1e-12
MLP test loss: 0.034754902733875395
LSTM test Loss: 0.004
```

Εικόνα 5.2: Σφάλματα μεθόδων πρόβλεψης για τη 2^η πειραματική ανάλυση

Σύμφωνα με τα αποτελέσματα της εικόνας 5.2, παρατηρείται ότι το Νευρωνικό Δίκτυο LSTM, αποδίδει καλύτερα στην εκτίμηση των μελλοντικών προβλέψεων των δεικτών IRI, συγκριτικά με τις γραμμικές συναρτήσεις.

Αναφορικά με τις γραμμικές μεθόδους πρόβλεψης, το σφάλμα των τριών γραμμικών συναρτήσεων είναι περίπου ίδιο, καθώς στις συναρτήσεις Ridge και Lasso Regression, η βέλτιστη παράμετρος «alpha» είναι ίση με 10^{-12} , το οποίο είναι πρακτικά ίσο με μηδέν. Έτσι μηδενίζονται (σχεδόν) οι παράμετροι που προστίθενται στον τύπο του σφάλματος των συναρτήσεων Ridge και Lasso Regression, με αποτέλεσμα το σφάλμα να είναι ίσο με αυτό της Linear Regression.

Ως βέλτιστη μέθοδος πρόβλεψης της 2^{ης} πειραματικής ανάλυσης, αναδείχθηκε αυτή του Νευρωνικού Δικτύου LSTM, της οποίας τα αποτελέσματα απεικονίζονται στο διάγραμμα του σχήματος 5.2.



Σχήμα 5.2: Πρόβλεψη των δεικτών IRI της δεξιάς λωρίδας του τμήματος Α και σύγκριση με τους πραγματικούς δείκτες για το ίδιο έτος – 2^η πειραματική ανάλυση

Σύμφωνα και με το Σχήμα 5.2, οι προβλέψεις του δικτύου δεν αποκλίνουν σημαντικά από τις πραγματικές τιμές των δεικτών. Επομένως, με την ολοκλήρωση της δεύτερης πειραματικής ανάλυσης, επαληθεύεται το συμπέρασμα της πρώτης

και αναδεικνύεται η ικανότητα του Νευρωνικού Δικτύου LSTM, ως προς την πρόβλεψη της μελλοντικής ομαλότητας του οδοστρώματος μίας οδού.

5.3 Αποτελέσματα 3^{ης} πειραματικής ανάλυσης

Το σφάλμα κάθε μεθόδου, για την εκτέλεση της 3^{ης} πειραματικής ανάλυσης, απεικονίζεται στην Εικόνα 5.3.

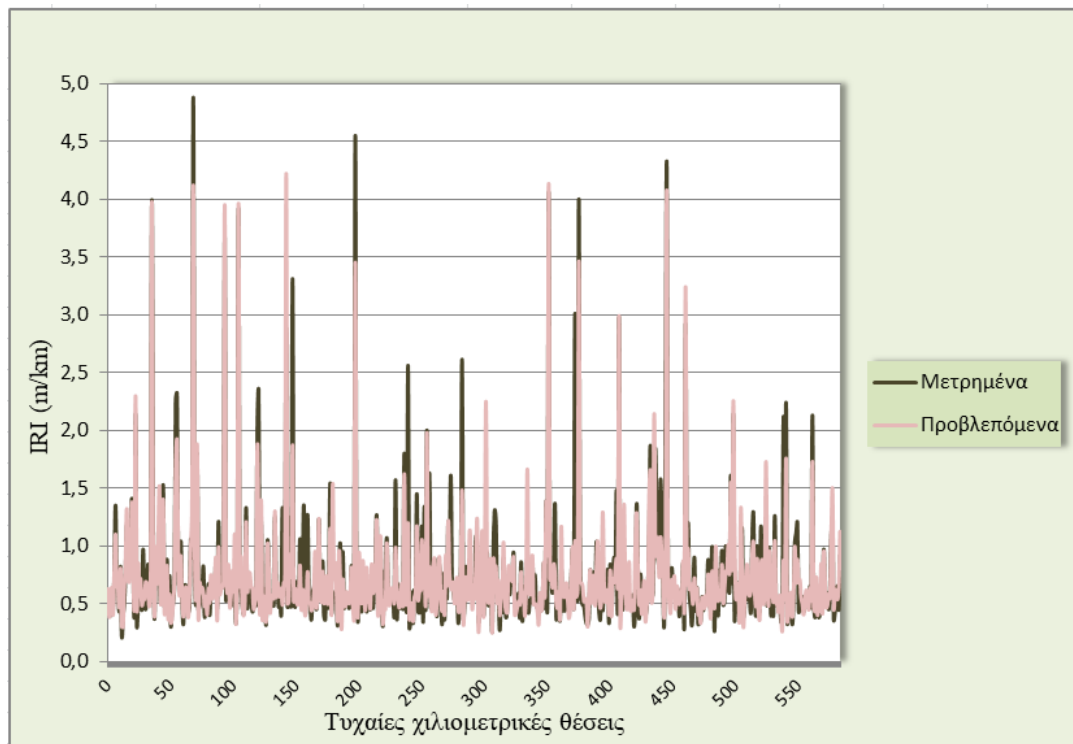
```
Error for LinearRegression is: 0.030914645953833683  
Smallest Error for Ridge is: 0.0309146459538337 with alpha: 1e-12  
Smallest Error for Lasso is: 0.030914645953833683 with alpha: 1e-12  
MLP test loss: 0.03062747221905049  
LSTM test Loss: 0.001
```

Εικόνα 5.3: Σφάλματα μεθόδων πρόβλεψης για την 3^η πειραματική ανάλυση

Σύμφωνα με τα αποτελέσματα της εικόνας 5.3, παρατηρείται ότι τα Νευρωνικά Δίκτυα MLP και LSTM, αποδίδουν καλύτερα στην εκτίμηση των μελλοντικών προβλέψεων των δεικτών IRI, συγκριτικά με τις γραμμικές συναρτήσεις.

Επιπλέον, όμοια με το πείραμα 2, το σφάλμα των τριών γραμμικών συναρτήσεων είναι περίπου ίδιο, καθώς στις συναρτήσεις Ridge και Lasso Regression, η βέλτιστη παράμετρος «alpha» είναι ίση με 10^{-12} , το οποίο είναι πρακτικά ίσο με μηδέν. Έτσι μηδενίζονται (σχεδόν) οι παράμετροι που προστίθενται στον τύπο του σφάλματος των συναρτήσεων Ridge και Lasso Regression, με αποτέλεσμα το σφάλμα να είναι ίσο με αυτό της Linear Regression.

Ως η βέλτιστη μέθοδος πρόβλεψης της 3^{ης} πειραματικής ανάλυσης, αναδείχθηκε αυτή του Νευρωνικού Δικτύου LSTM, της οποίας τα αποτελέσματα απεικονίζονται στο διάγραμμα του σχήματος 5.3.



Σχήμα 5.3: Πρόβλεψη των δεικτών IRI της αριστερής λωρίδας του τμήματος Β και σύγκριση με τους πραγματικούς δείκτες για το ίδιο έτος – 3^η πειραματική ανάλυση

Σύμφωνα και με το Σχήμα 5.3, οι προβλέψεις του δικτύου δεν αποκλίνουν σημαντικά από τις πραγματικές τιμές των δεικτών. Επομένως, είναι δυνατή η πρόβλεψη της μελλοντικής ομαλότητας του οδοστρώματος του τμήματος μίας οδού, μέσω ενός δικτύου LSTM εκπαιδευμένου με στοιχεία ενός άλλου τμήματος, τα οποία δε διαφέρουν κατά πολύ με αυτά του προβλεπομένου.

5.4 Αποτελέσματα 4^{ης} πειραματικής ανάλυσης

Το σφάλμα κάθε μεθόδου, για την εκτέλεση της 4^{ης} πειραματικής ανάλυσης, απεικονίζεται στην Εικόνα 5.4.

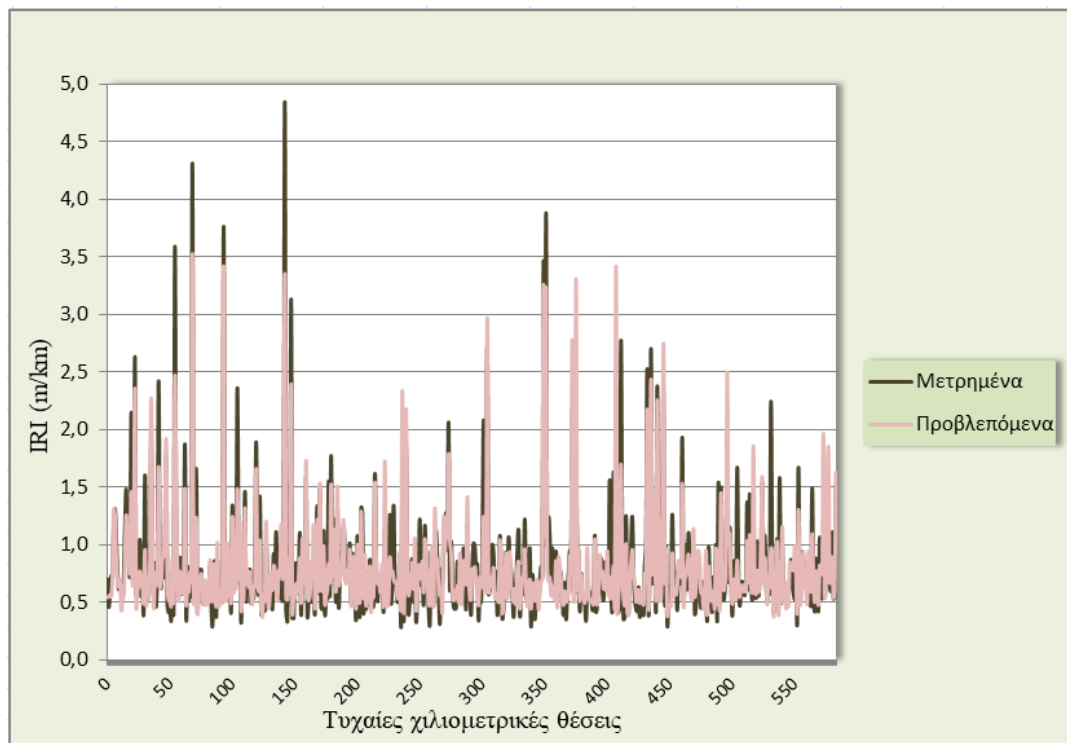
```
Error for LinearRegression is: 0.037255759493324256
Smallest Error for Ridge is: 0.03725817430973205 with alpha: 1e-12
Smallest Error for Lasso is: 0.03725817430973224 with alpha: 1e-12
MLP test loss: 0.0393333208063859
LSTM test Loss: 0.002
```

Εικόνα 5.4: Σφάλματα μεθόδων πρόβλεψης για την 4^η πειραματική ανάλυση

Σύμφωνα με τα αποτελέσματα της εικόνας 5.2, παρατηρείται ότι το Νευρωνικό Δίκτυο LSTM, αποδίδει καλύτερα στην εκτίμηση των μελλοντικών προβλέψεων των δεικτών IRI, συγκριτικά με τις γραμμικές συναρτήσεις.

Όπως και στα πειράματα 2 και 3, το σφάλμα των τριών γραμμικών συναρτήσεων είναι πρακτικά ίσο, λόγω της μικρής τιμής της παραμέτρου «alpha», η οποία μηδενίζει (σχεδόν) τις παραμέτρους που προστίθενται στους τύπους των σφαλμάτων των συναρτήσεων Ridge και Lasso Regression. Έτσι, το σφάλμα των Ridge και Lasso Regression είναι περίπου ίσο με αυτό της Linear Regression.

Για ακόμα μία φορά, βέλτιστη μέθοδος πρόβλεψης της 4^{ης} πειραματικής ανάλυσης, αναδείχθηκε αυτή του Νευρωνικού Δικτύου LSTM, της οποίας τα αποτελέσματα απεικονίζονται στο διάγραμμα του σχήματος 5.4.



Σχήμα 5.4: Πρόβλεψη των δεικτών IRI της δεξιάς λωρίδας του τμήματος Β και σύγκριση με τους πραγματικούς δείκτες για το ίδιο έτος – 4^η πειραματική ανάλυση

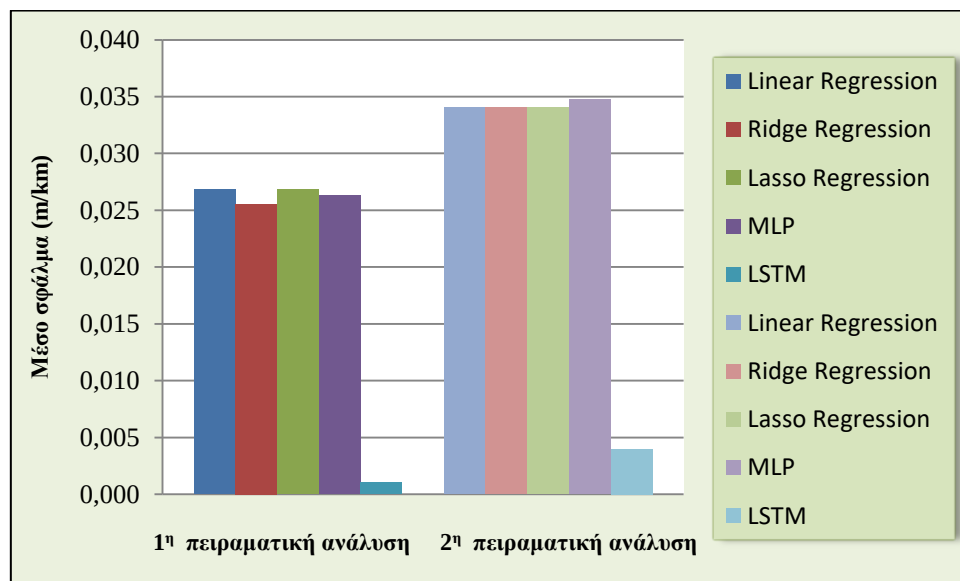
Σύμφωνα και με το Σχήμα 5.4, οι προβλέψεις του δικτύου δεν αποκλίνουν σημαντικά από τις πραγματικές τιμές των δεικτών. Επομένως, με την ολοκλήρωση της τέταρτης πειραματικής ανάλυσης, επαληθεύεται το συμπέρασμα της τρίτης και αναδεικνύεται η ικανότητα γενίκευσης του Νευρωνικού Δικτύου LSTM, ως προς την πρόβλεψη της μελλοντικής ομαλότητας του οδοστρώματος του τμήματος μίας

οδού, μέσω ενός δικτύου LSTM εκπαιδευμένου με στοιχεία ενός άλλου τμήματος, τα οποία δε διαφέρουν κατά πολύ με αυτά του προβλεπομένου.

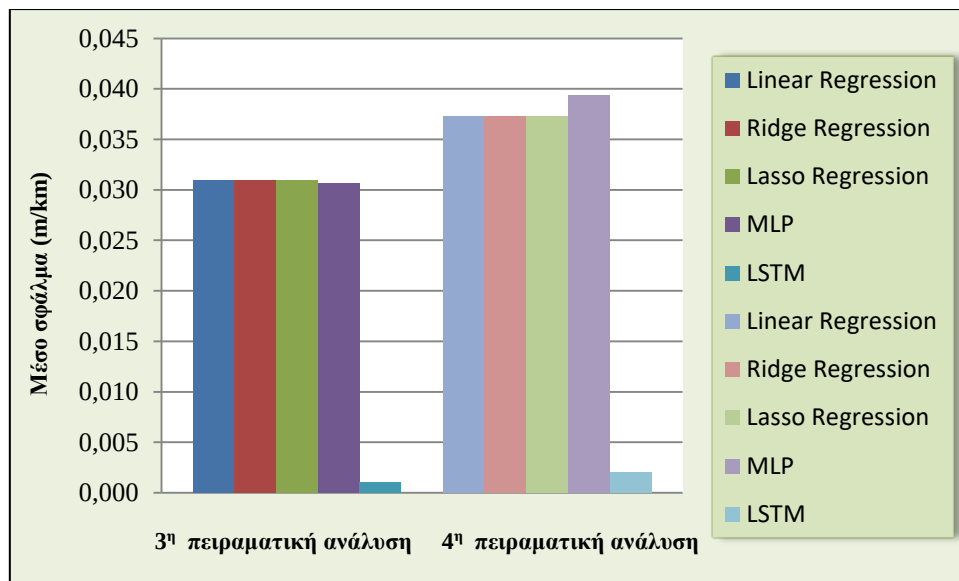
5.5 Συγκριτική ανάλυση

Οι πειραματικές αναλύσεις 2 και 4 πραγματοποιούνται για την επαλήθευση των αποτελεσμάτων που προέκυψαν από τις αναλύσεις 1 και 3 αντίστοιχα. Για αυτόν το λόγο, αξίζει να σημειωθεί και να σχολιασθεί η συγκριτική ανάλυση μεταξύ της 1^{ης} και της 2^{ης} πειραματικής ανάλυσης, αλλά και η συγκριτική ανάλυση μεταξύ της 3^{ης} και της 4^{ης} πειραματικής ανάλυσης.

Στα σχήματα 5.5 και 5.6, απεικονίζεται η συγκριτική ανάλυση μεταξύ των πειραματικών αναλύσεων 1-2 και 3-4 αντίστοιχα.



Σχήμα 5.5: Συγκριτική ανάλυση 1^{ης} και 2^{ης} πειραματικής ανάλυσης



Σχήμα 5.6: Συγκριτική ανάλυση 3^{ης} και 4^{ης} πειραματικής ανάλυσης

Από τη συγκριτική ανάλυση των δύο πρώτων πειραματικών αναλύσεων, παρατηρείται ότι τα σφάλματα των μεθόδων πρόβλεψης είναι μεγαλύτερα στη 2^η σε σχέση με την 1^η. Η απόκλιση είναι περίπου 0,07 στις μεθόδους πρόβλεψης μέσω γραμμικών συναρτήσεων, αλλά και στη μέθοδο πρόβλεψης μέσω του δικτύου MLP. Ομοίως, το σφάλμα του Νευρωνικού Δικτύου LSTM, είναι μεγαλύτερο στη 2^η πειραματική ανάλυση, κατά 0,03. Σημειώνεται ότι, η διαφορά σφάλματος μεταξύ των δύο αυτών πειραματικών αναλύσεων δε χαρακτηρίζεται σημαντική και επομένως και στις δύο αυτές αναλύσεις τα αποτελέσματα που προκύπτουν δεν απέχουν πολύ από τα πραγματικά. Οι παρατηρήσεις από τη δεύτερη συγκριτική ανάλυση, των πειραματικών αναλύσεων 3 και 4, είναι όμοιες με τις προαναφερθείσες.

Από την άλλη, η διαφορά αυτή των σφαλμάτων, μπορεί να οφείλεται στις κατά περίπτωση υψηλές τιμές του δείκτη IRI, οι οποίες εμφανίζονται πιο συχνά στις μετρήσεις της αριστερής λωρίδας (2^η & 4^η πειραματική ανάλυση). Ακόμα, παρατηρείται ότι το σφάλμα του Νευρωνικού Δικτύου MLP είναι λίγο μεγαλύτερο από τα σφάλματα των γραμμικών συναρτήσεων, χωρίς να έχει σημαντική διαφορά. Αυτό οφείλεται στο γεγονός ότι τα στοιχεία αξιοποίησης για τις πειραματικές αναλύσεις, δηλαδή τα πάχη του οδοστρώματος και τα στοιχεία κυκλοφορίας, έχουν άμεση συσχέτιση με τους δείκτες IRI. Επομένως, τα σφάλματα που προκύπτουν από τις αναλύσεις μέσω γραμμικών συναρτήσεων, δε διαφέρουν σημαντικά με τα σφάλματα που προκύπτουν από τις αναλύσεις μέσω του δικτύου MLP. Τέλος, σε κάθε πειραματική ανάλυση, το σφάλμα που προκύπτει από το Νευρωνικό Δίκτυο

LSTM, είναι αρκετά μικρότερο από αυτά των άλλων μεθόδων πρόβλεψης, καθώς το δίκτυο LSTM, λόγω της χρονικής αντίληψής του, αποδίδει καλύτερα στην πρόβλεψη των μελλοντικών δεικτών IRI.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, αναδείχθηκε η ικανότητα των Νευρωνικών Δικτύων να προβλέπουν την ομαλότητα του οδοστρώματος μίας οδού, μέσω της πρόβλεψης τιμών του διεθνή δείκτη ομαλότητας IRI. Από τη βιβλιογραφική ανασκόπηση και την πραγματοποίηση των πειραματικών αναλύσεων, προέκυψαν τα ακόλουθα συμπεράσματα:

- Μεταξύ των υπό διερεύνηση μεθόδων, τα Νευρωνικά Δίκτυα LSTM (Long Short Term Memory Networks) αναδεικνύονται ως η καλύτερη μέθοδος πρόβλεψης της ομαλότητας των οδοστρωμάτων.
- Η πρόβλεψη της ομαλότητας του οδοστρώματος ενός τμήματος μίας οδού, μπορεί να πραγματοποιηθεί με χρήση ενός Νευρωνικού Δικτύου, το οποίο έχει εκπαιδευτεί για ένα άλλο τμήμα. Βασική προϋπόθεση είναι το τμήμα αυτό να έχει ίδιο τύπο ασφατικής στρώσης κυκλοφορίας και να παρουσιάζει ανάλογα στοιχεία κυκλοφορίας, παχών οδοστρώματος και κατά μήκος ομαλότητα, καθώς και ίδιες καιρικές και περιβαλλοντικές συνθήκες.
- Σε πειραματικές αναλύσεις, όπου τα δεδομένα εισόδου συνδέονται άμεσα με τη ζητούμενη πρόβλεψη, όπως και στην παρούσα διπλωματική όπου τα πάχη του οδοστρώματος και τα στοιχεία κυκλοφορίας συνδέονται άμεσα με την πρόβλεψη των δεικτών IRI, οι γραμμικές συναρτήσεις φαίνεται ότι δίνουν συναφή αποτελέσματα με το Νευρωνικό Δίκτυο MLP.
- Όπως και σε κάθε άλλη πειραματική ανάλυση μελλοντικών προβλέψεων, έτσι και οι προβλέψεις που προκύπτουν από ένα Νευρωνικό Δίκτυο δεν είναι 100% έγκυρες. Η ομαλότητα ενός οδοστρώματος μεταβάλλεται χρονικά και εξαρτάται από τη μεταβολή της κυκλοφορίας της οδού, αλλά και τις καιρικές και περιβαλλοντικές συνθήκες που θα επικρατήσουν σε μελλοντικό χρόνο.
- Για την καλύτερη εκπαίδευση ενός Νευρωνικού Δικτύου, με στόχο την ελαχιστοποίηση του σφάλματος όσον αφορά στην πρόβλεψη της ομαλότητας των οδοστρωμάτων, είναι επιθυμητή η άντληση όσο το δυνατόν περισσότερων δεδομένων για το τμήμα μίας οδού, από προηγούμενα έτη.

Εν κατακλείδι, τα Νευρωνικά Δίκτυα, μέσω των προβλέψεων των μελλοντικών τιμών του δείκτη IRI του οδοστρώματος ενός τμήματος μίας οδού, δίνουν λύση σε προβλήματα μηχανικού, όπως είναι ο έγκαιρος προγραμματισμός και η λήψη μέτρων συντήρησης του οδοστρώματος με στόχο την ελαχιστοποίηση του χρόνου επέμβασης αλλά και του κόστους.

Αντίστοιχη διερεύνηση θα μπορούσε να γίνει για την πρόβλεψη της εξέλιξης άλλων χαρακτηριστικών των οδοστρωμάτων, όπως είναι η επιφανειακή υφή, η ολισθηρότητα αλλά και η ρηγμάτωση του οδοστρώματος. Σε κάθε περίπτωση, τα Νευρωνικά Δίκτυα αποτελούν ένα υποβοηθητικό εργαλείο για τον μηχανικό, καθώς η λήψη αποφάσεων στο πλαίσιο ενός συστήματος διαχείρισης των οδοστρωμάτων βασίζεται στον συγκερασμό του συνόλου των παραμέτρων που υπεισέρχονται στην αξιολόγηση της κατάστασης των οδοστρωμάτων.

7. Βιβλιογραφία

- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., Linus, O. U., et al. (2019). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Journals & Magazine*, 7, pp. 158820-158846.
- Alam, B. (2022, June 4). *Hands-On.Cloud*. Retrieved February 2023, from Simple Sklearn Ridge Regression Example In Python: https://hands-on.cloud/sklearn-ridge-regression-example/?utm_content=cmp-true
- Bhandare, A., & Kaur, D. (2021). Designing Convolutional Neural Network Architecture Using Genetic Algorithms. *International Journal of Advanced Network, Monitoring and Controls*, 6, pp. 26-35.
- Chen, S.-L., Lin, C.-H., Tang, C.-W., Chu, L.-P., & Cheng, C.-K. (2020). Research on the International Roughness Index Threshold of Road Rehabilitation in Metropolitan Areas: A Case Study in Taipei City. *Sustainability*, 12(24), pp. 10536-10555.
- Devore, J., Hossain, M., & William H. Parcells, J. (1995). Automated System for Pavement Profile Analysis from Profilograph Traces. *Transportation Research Record: Journal of Transportation Research Board*, 1505, pp. 47-56.
- Domashova, J. V., Emtseva, S. S., Fail, V. S., & Gridin, A. S. (2021). Selecting an optimal architecture of neural network using genetic. *Procedia Computer Science: 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society*, 190, pp. 263-273.
- Domitrovic, J., Dragovan, H., Rukavina, T., & Dimter, S. (2018). Application of an Artificial Neural Network in Pavement Management System. *Tehnicki Vjesnik*, 25, pp. 466-473.
- El-Hakim, R. A., & El-Badawy, S. (2013). International Roughness Index Prediction for Rigid Pavements: An Artificial Neural Network Application. *Advanced Materials Research*, 723, pp. 854-860.
- Georgiou, P., Plati, C., & Loizos, A. (2015). Pavement Roughness Progression Modeling using Artificial Neural Networks and Multivariate Adaptive Regression Splines. *Civil-Comp Proceedings: Proceedings of the Fourth International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, 109.
- Georgiou, P., Plati, C., & Loizos, A. (2018). Soft Computing Models to Predict Pavement Roughness: A Comparative Study. *Advances in Civil Engineering*, 2018.
- Gurney, K. (1997). What are neural networks? In K. Gurney, *An introduction to neural networks* (4 ed., Vol. 40, pp. 13-16). London: UCL Press.

- HP, S. (2019, July 27). *Linear Regression in Statistics and Machine learning*. Retrieved February 2023, from MLearning.ai: <https://medium.com/mllearning-ai/linear-regression-in-statistics-and-machine-learning-5407b7de7a9e>
- Islam, S., Buttlar, W. G., Aldunate, R. G., & Vavrik, W. R. (2015). Measurement of Pavement Roughness Using Android-Based Smartphone Application. *Transportation Research Record: Journal of the Transportation Research Board*, 2457(1), pp. 30-38.
- Krogh, A. (2008). What are artificial neural networks? *Nature Biotechnology*, 26(2), pp. 195-197.
- Macukow, B. (2016). Neural Networks – State of Art, Brief History, Basic Models and Architecture. *Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference*, 9842, pp. 3-14.
- Maind, S. B., & Wankar, P. (2014). Research Paper on Basic of Artificial Neural Network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1), pp. 96-100.
- Marini, F., Bucci, R., Magrì, A., & Magrì, A. (2008). Artificial neural networks in chemometrics: History, examples and perspectives. *Microchemical Journal*, 88(2), pp. 178-185.
- Mazari, M., & D.Rodriguez, D. (2016). Prediction of pavement roughness using a hybrid gene expression programming-neural network technique. *Journal of traffic and transportation engineering (english edition)*, 3(5), pp. 448-455.
- Mondal, A., Hand, A. J., & Ward, D. R. (2000). *Evaluation of Lightweight Non-Contact Profilors*. West Lafayette, Indiana.
- MÚČKA, P. (2012). Relationship between International Roughness Index and Straightedge Index. *Journal of Transportation Engineering*, 138(9), pp. 1099-1112.
- Nazef, A., Mraz, A., Iyer, S., & Choubane, B. (2009). A Semi-Automated Faulting Measurement Approach For Rigid Pavements Using High Speed Inertial Profiler Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2094(1), pp. 121-127.
- Perera, R., Byrum, C., & Kohn, S. (1998). *Investigation of Development of Pavement Roughness*. U.S. Department of Transportation-Federal Highway Administration.
- Perlato, A. (2019, June 16). AP. Retrieved February 2023, from <https://www.andreaperlato.com/theorypost/ridge-and-lasso-regression/>
- Pienaar, S. W., & Malekian, R. (2019). Human Activity Recognition Using LSTM-RNN Deep Neural Network Architecture. *2019 IEEE 2nd Wireless Africa Conference*, pp. 1-5.

- Ruby, D., Theerthagiri, P., Jacob, D., & Dr.Y.Vamsidhar. (2020). Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(4), pp. 5393-5397.
- Sak, H., Senior, A., & Beaufays, F. (2014). Long Short-Term Memory Recurrent Neural Network Architectures. *Interspeech*, pp. 338-342.
- Sayers, M. W., & Karamihas, S. M. (1996). *Interpretation of Road Roughness Profile Data*. Michigan: The University of Michigan Transportation Research Institute.
- Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228, pp. 2313-2324.
- Smith, K. A., & Gupta, J. N. (2000). Neural networks in business: techniques and applications for the operations researcher. *Computers & Operations Research*, 27(11), pp. 1023-1044.
- Spangler, E. B., & Kelly, W. J. (1994). Development and Evaluation of the Ride Number Concept. In B. T. Kulakowski, A. International, & V.-R. I. Conference, *Vehicle -Road Interaction* (pp. 135-149). Philadelphia: ASTM.
- (2015). Advantages of Neural Network Method for Solving Different Equations. In N. Yadav, A. Yadav, & M. Kumar, *An Introduction to Neural Network Methods for Differential Equations* (pp. 11-12). Springer.
- Yousefzadeh, M., Azadi, S., & Soltani, A. (2009). Road profile estimation using neural network algorithm. *Journal of Mechanical Science and Technology*, 24(3), pp. 743-754.
- Λοΐζος, Α., & Πλατή, Χ. (2021, February). In Α. Λοΐζος, & Χ. Πλατή, *Αξιολόγηση και Συντήρηση Οδοστρωμάτων* (pp. 116-122). Αθήνα: ΕΜΠ.